

**IMPLEMENTACIÓN DE UN SERVICIO TELEMÁTICO SOBRE LA RED
MULTISERVICIOS DE EMCALI**

ANDRÉS FELIPE MURILLO PIEDRAHITA

**UNIVERSIDAD AUTÓNOMA DE OCCIDENTE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE AUTOMÁTICA Y ELECTRÓNICA
PROGRAMA INGENIERÍA ELECTRÓNICA
SANTIAGO DE CALI**

2009

**IMPLEMENTACIÓN DE UN SERVICIO TELEMÁTICO SOBRE LA RED
MULTISERVICIOS DE EMCALI**

ANDRÉS FELIPE MURILLO PIEDRAHITA

Pasantía en investigación para optar al título de ingeniero electrónico

Director

OSCAR HERNÁN MONDRAGÓN MARTINEZ
Ingeniero en electrónica y telecomunicaciones

UNIVERSIDAD AUTÓNOMA DE OCCIDENTE

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE AUTOMÁTICA Y ELECTRÓNICA

PROGRAMA DE INGENIERÍA ELECTRÓNICA

SANTIAGO DE CALI

2009

Nota de aceptación:

**Aprobado por el Comité de
Grado en cumplimiento
con los requisitos
exigidos por la
Universidad Autónoma de
Occidente para optar al
título de Ingeniero
Electrónico**

**OSCAR HERNÁN
MONDRAGON**

Director

Helmut Rubio

Jurado

Jaír Abadía Correa

Jurado

Santiago de Cali, 18 de Mayo de 2009

Dedico este trabajo a mi madre, apoyo fundamental en mi proceso de formación como profesional y ejemplo a seguir por su perseverancia. También lo dedico a mis amigos y compañeros de estudio que me acompañaron a lo largo de este recorrido, su apoyo fue indispensable en algunos momentos de duda.

AGRADECIMIENTO

Extiendo mis agradecimientos a las empresas públicas de Cali, EMCALI por permitir el desarrollo de este proyecto y abrir las puertas para utilizar la infraestructura de su red multiservicios.

También agradezco al director de proyecto de grado, Oscar Hernán Mondragón y directora de Programa, Zeida María Solarte. Su apoyo fue indispensable en los momentos de dificultad.

CONTENIDO

GLOSARIO.....	14
RESUMEN.....	18
INTRODUCCIÓN.....	19
1.DESCRIPCIÓN GENERAL DEL PROYECTO.....	21
1.1 TÍTULO DEL PROYECTO.....	21
1.2 PARTICIPANTES.....	21
1.3 DEFINICIÓN DEL PROBLEMA.....	21
1.4 OBJETIVOS.....	22
1.5 MARCO TEÓRICO.....	22
1.6 ANTECEDENTES.....	30
1.7 JUSTIFICACIÓN.....	33
2.REDES DE NUEVA GENERACIÓN (NGN).....	35
2.1 HISTORIA DE LAS REDES DE NUEVA GENERACIÓN.....	35
2.2 PLATAFORMAS NGN.....	38
2.3 RED MULTISERVICIOS DE EMCALI.....	41
3.HERRAMIENTAS PARA LA CREACIÓN DE NUEVAS APLICACIONES EN NGN.....	48
3.1 SIP SERVLETS.....	48
3.2 CCXML (CALL CONTROL EXTENSIBLE MARKUP LANGUAGE).....	48
3.3 JAVA TELEPHONY API (JTAPI).....	50
3.4 JAIN CALL CONTROL (JCC / JCAT).....	55
3.5 SERVICE CREATION MARKUP LANGUAGE (SCML).....	58
3.6 OSA/PARLAY.....	60

4.OSA/PARLAY.....	69
4.1 HISTORIA DE OSA/PARLAY.....	70
4.2 MODELO CONCEPTUAL DE OSA/PARLAY.....	71
4.3 PATRONES DE DISEÑO Y CONVENCIONES DE NOMBRES.....	74
4.4 CAPACIDADES DE SERVICIO EN OSA/PARLAY.....	78
4.5 ACCESO A LOS SERVICIOS Y LAS APLICACIONES.....	86
4.6 MANTENIMIENTO DE LOS SERVICIOS.....	94
5.DISEÑO DEL SERVICIO.....	97
5.1 OBJETIVO DEL SOFTWARE.....	97
5.2 ALCANCE.....	97
5.3 DEFINICIÓN DEL SISTEMA.....	98
5.4 LISTA DE REQUERIMIENTOS FUNCIONALES.....	98
5.5 REQUERIMIENTOS NO FUNCIONALES.....	99
5.6 ADMINISTRACIÓN DE CARGAS Y FALTAS.....	99
5.7 DEFINICIÓN DE ACTORES.....	100
5.8 CASOS DE USO.....	100
5.9 MODELO DE FLUJO DE DATOS.....	107
5.10 DIAGRAMA DE ESTADOS.....	108
5.11 DIAGRAMA DE CLASES.....	110
6.IMPLEMENTACIÓN DEL SERVICIO.....	116
6.1 HERRAMIENTAS UTILIZADAS PARA LA IMPLEMENTACIÓN.....	116
6.2 PLAN DE PRUEBAS.....	117
6.3 RESULTADOS DE LAS PRUEBAS.....	122
7.RECOMENDACIONES AL MODELO ARQUITECTÓNICO PARA LA IMPLEMENTACIÓN DE NUEVOS SERVICIOS TELEMÁTICOS SOBRE LA RED MULTI-SERVICIOS.....	126
7.1 RECOMENDACIONES REALIZADAS AL NIVEL TRES.....	126
7.2 RECOMENDACIONES REALIZADAS AL NIVEL DOS.....	128

7.3 RECOMENDACIONES REALIZADAS AL NIVEL UNO.....	129
8.CONCLUSIONES.....	131
ANEXOS.....	134

LISTA DE FIGURAS

Figura 1. Arquitectura típica de una NGN.....	25
Figura 2. Modelo de servicio de Red Inteligente.....	37
Figura 3. Arquitectura de Red de Nueva Generación.....	40
Figura 4. Red Multiservicios de EMCALI.....	43
Figura 5. Contenedor de SIP Servlet.....	49
Figura 6. Modelo de aplicaciones en CCXML.....	50
Figura 7. Modelo de llamada de JTAPI.....	52
Figura 8. Implementación de JTAPI y Peers.....	53
Figura 9. Diagrama de secuencia del establecimiento de una llamada en JTAPI	54
Figura 10. Arquitectura básica JAIN.....	56
Figura 11. Modelo de objetos de JCAT para una llamada entre dos partes.....	57
Figura 12. Patrón de programación usando Java Listeners.....	58
Figura 13. Papel del intérprete en SCML.....	59
Figura 14. Creación, implementación y ejecución de un script.....	60
Figura 15. OSA/Parlay como Gateway en una red NGN.....	62
Figura 16. Relación entre Parlay X y los API's de Parlay.....	65
Figura 17. Triángulo de OSA/Parlay.....	72
Figura 18. Interacción entre una aplicación y la red.....	73
Figura 19. Interacción entre el SCS y la red.....	74
Figura 20. Framework como punto de contacto entre SCS y aplicaciones cliente	74
Figura 21. Funcionamiento métodos asíncronos.....	76
Figura 22. Funcionamiento métodos síncronos.....	76
Figura 23. Organización de notificaciones.....	78
Figura 24. Relación entre los tipos de Control de Llamadas.....	80
Figura 25. Ejemplo del uso de la SCF Interacción con Usuario.....	82
Figura 26. Acceso a la interfaz inicial.....	88
Figura 27. Autenticación ante el Framework.....	89
Figura 28. Diagrama de secuencia proceso de autenticación.....	90

Figura 29. Descubrimiento de servicios.....	91
Figura 30. Uso de la interfaz IpServiceDiscovery.....	91
Figura 32. Firma del acuerdo de servicio.....	92
Figura 33. Acuerdo mutuo de servicio.....	93
graphics2.....	94
Figura 34. Actores del servicio.....	100
Figura 35. Diagrama de casos de uso.....	107
Figura 36. Diagrama de flujo de datos servicio de Portabilidad Numérica.....	108
Figura 37. Diagrama de estados.....	109
Figura 38. Diagrama de clases servicio Portabilidad Numérica.....	112
Figura 39. Diagrama de secuencia servicio de Portabilidad Numérica.....	114
Figura 40. Diagrama de flujo de pruebas realizadas.....	120
Figura 41 Pruebas sin realizar.....	122
Figura 42. Ejecución de la aplicación.....	123
Figura 43. Funcionamiento sistema de logs.....	123
Figura 44. Inicialización exitosa de CgasClientHub.....	123
Figura 45 Registro de la aplicación.....	124
Figura 46. Acceso a servicio GCC.....	124
Figura 47. OAM con opciones no habilitadas.....	125
Figura 48. Diagrama de clases Nivel 3.....	127
Figura 49. Diagrama de secuencia nivel 3 Diseño de servicio de prueba	128
Figura 50. Diagrama de clases Nivel 2.	129
Figura 51. Diagrama de secuencia Nivel 2. Diseño de servicio de prueba.....	129
Figura 52. Diagrama de clases Nivel 1.....	130
Figura 53. Diagrama de secuencia nivel 1. Diseño servicio de prueba.....	130

LISTA DE TABLAS

Tabla 1. Capacidades de Servicio en OSA/Parlay.....	63
Tabla 2. Flujo de eventos Caso de Uso Enrutamiento.....	101
Tabla 3. Flujo de eventos Caso de Uso CUA1.....	102
Tabla 4. Flujo de eventos Caso de Uso CUA1.1.....	102
Tabla 5. Flujo de eventos Caso de Uso CUA1.2.....	103
Tabla 6. Flujo de eventos Caso de Uso CUA2.....	104
Tabla 7. Flujo de eventos Caso de Uso CUA2.1.....	104
Tabla 8. Flujo de eventos Caso de Uso CUA2.2.....	105
Tabla 9. Flujo de eventos Caso de Uso CUA2.2.1.....	106
Tabla 10: Prueba 1. Ejecución del código.....	117
Tabla 11. Prueba 2. Sistema de logs.....	117
Tabla 12. Prueba 3. Inicialización de CgasClientHub.....	117
Tabla 13. Prueba 4 Registro de aplicaciones en OAM.....	118
Tabla 14. Prueba 5. Autorización y acceso a servicios.....	118
Tabla 15. Prueba 6. Creación de notificación.....	118
Tabla 16. Prueba 7. Recepción de notificación.....	118
Tabla 17. Prueba 8. Direccionamiento.....	119

LISTA DE ANEXOS

Anexo A. Acta de reunión Octubre 3 de 2008.	134
Anexo B. Acta de reunión Marzo 9 de 2009.....	135
Anexo C. Acta de reunión Marzo 11 de 2009.....	136
Anexo D. Acta de reunión Abril 17 de 2009.....	137
Anexo E. Acta de reunión Marzo 12 de 2009.....	138
Anexo F . Modelo para el desarrollo de servicios en la red NGN de emcali – telecomunicaciones.....	139

GLOSARIO

3GPP (3rd Group Partnership Project): se trata de un consorcio colaborativo entre asociaciones de telecomunicaciones para establecer una especificación para sistemas móviles que sea globalmente aplicable. OSA/Parlay es actualmente sostenido y actualizado por este grupo.

CARRIER-CLASS: es un servicio prestado con el 99.999% de disponibilidad.

CHAP: es protocolo de autenticación. Se utiliza al iniciar un enlace y verifica de forma periódica la identidad del cliente remoto usando un intercambio de información de tres etapas. Chap se realiza al establecer el enlace inicial y se repite durante el tiempo que dure el enlace.

CORBA: es un estándar definido por el Object Management Group que permite escribir componentes de software en múltiples lenguajes y ejecutarlos en múltiples computadoras trabajando cooperativamente. Es una arquitectura estándar para sistemas de objetos distribuidos. El estándar define las API's, el protocolo de comunicaciones y los mecanismos para asegurar la interoperabilidad.

CPE: equipo terminal del abonado. Son usados para originar, encaminar o terminar una comunicación. Estos dispositivos se encuentran localizados del lado del cliente y se encuentran conectados con el canal de comunicaciones del proveedor.

DSL (DIGITAL LINE SUSCRIBER): línea de abonado digital. Es una tecnología de banda ancha que utiliza líneas telefónicas de par trenzado para transportar datos de alto ancho de banda para dar servicio a los suscriptores. La tecnología DSL permite que se puedan realizar conexiones telefónicas y de datos al tiempo, permite cargar y descargar datos a frecuencias superiores a 4KHz.

FRAME RELAY: es un protocolo de capa de enlace de datos en el modelo OSI. Funciona a través de la retransmisión de tramas. La velocidad de datos disponible es de por lo menos 4Mbps. No realiza ningún control de flujo ni errores. Las conexiones Frame Relay utilizan circuitos virtuales.

GATEWAY: es un elemento que permite la conexión entre dos redes diferentes o redes que usan distintos protocolos.

GSM (Global System for Mobile Communications): es el estándar más popular para teléfonos móviles en el mundo. Es considerado un sistema de

telefonía móvil de segunda generación.

IAD (DISPOSITIVO DE ACCESO INTEGRADO): es un dispositivo instalado en las premisas del cliente para proveer el acceso a redes de Internet. Agrega múltiples canales de información incluyendo voz y datos a través de un único enlace de acceso compartido hacia el proveedor del servicio.

IDL (Interface Definition Language): es un lenguaje de definición de interfaces que se utiliza en sistemas de programación distribuidos. Ofrece sintaxis para definir los objetos y métodos que requieren invocar de forma remota. Este lenguaje es multiplataforma.

IMAP (Internet Message Access Protocol): es un protocolo de red de acceso a mensajes electrónicos almacenados en un servidor.

MGCP (Media Gateway Control Protocol): es un protocolo de control de dispositivos, donde un Gateway esclavo, es controlado por un maestro MGC (Media Gateway Controller). Está basado en redes de telefonía IP.

MPLS (MultiProtocol Label switching): es un mecanismo de transporte de datos estándar creado por la IETF, opera entre las capas de enlace de datos y la capa de red. Puede ser utilizado para transportar voz y paquetes IP, esto lo hace a través de la creación de circuitos virtuales. Mientras viajan a través de estos circuitos los paquetes son etiquetados según prioridad y calidad. La conmutación de paquetes usa como criterio las etiquetas añadidas.

MULTICAST: es un servicio de red donde el flujo de datos de una fuente puede ser enviado de forma simultánea a varios destinatarios.

NAS (Network Attached Storage): es una tecnología de almacenamiento dedicada a compartir la capacidad de almacenamiento de un servidor con ordenadores personales u otros servidores cliente a través de una red. El flujo de datos es controlado por TCP/IP.

OSA(Open Service Access): este proyecto del 3gpp define un grupo de interfaces abiertas para el desarrollo de nuevos servicios en telecomunicaciones, el API de OSA es conocido como Parlay.

POP (Post Office Protocol): es un estándar perteneciente a la capa de aplicación en el modelo de Internet. Es utilizado por los clientes de correo para acceder a correos de un servidor remoto.

POTS (Plain Old Telephone Service) y PSTN (Public Switched Telephone Network):

estos términos definen los servicios telefónicos analógicos que utilizan hilos de cobre. Está constituido por todos los medios de transmisión y conmutación necesario para establecer una comunicaciones entre dos terminales. Trabaja

entre las frecuencias de 300Hz y 3400Hz, tiene un ancho de banda de 56Kbps.

PPP (Point to Point Protocol): protocolo punto a punto. Es un protocolo usado por las redes WAN conmutadas. Permite establecer una comunicación a nivel de enlace de datos entre dos equipos. Es utilizado en conexiones de internet y facilita la autenticación por medio de claves de acceso.

PROTOCOLO SIP (Session Initiation Protocol): es un protocolo que opera en la capa de aplicación y es empleado para controlar llamadas multimedia y servicios telefónicos avanzados. Está basado en http y es empleado en Internet. Los usuarios se identifican por medio de direcciones que se asemejan a las usadas en los correos electrónicos. Cuenta con funciones para localizar a los usuarios durante las llamadas y permite negociar el nivel de servicios ofrecidos durante una llamada según las capacidades de los terminales.

RUP(Ration Unified Proccess): es un proceso iterativo de desarrollo de software. El desarrollo consiste en una serie de iteraciones que van añadiendo funciones al sistema hasta obtener un producto final. Al final de cada iteración se realiza una retroalimentación según los resultados obtenidos.

SCF (Service Capability Function): es una interfaz que encapsula una funcionalidad determinada de la red. Estas funcionalidades pueden ir desde control de llamadas, hasta aseguramiento de calidad de servicio.

SCS (Service Capability Server): es una entidad que ha implementado un SCF y lo ofrece para las aplicaciones.

SMPP (Short Message Peer to Peer Protocol): es un protocolo estándar utilizado para enviar Mensajes de Texto Cortos (SMS). El protocolo se basa en el intercambio petición/respuesta, el intercambio de datos puede ser síncrono o asíncrono.

SMTP (Simple Mail Transfer Protocol): protocolo simple de transferencia de correo. Es un protocolo basado en texto utilizado para intercambiar mensajes de correo electrónico. Se basa en el modelo cliente/servidor.

SNMP (Simle Network Management Protocol): es un protocolo simple utilizado en la gestión de redes. Permite visualizar el estado de los elementos de la red. Estas variables pueden ser solicitadas periódicamente o por orden de la entidad gestionadora.

SOAP (Simple Object Access Protocol): es un protocolo que define como dos objetos de diferentes procesos pueden comunicarse por medio del intercambio de datos XML. Es utilizado en los servicios Web. No se encuentra asociado a ningún lenguaje ni a ningún protocolo de la capa de transporte.

SOFTSWITCH: es un agente de llamadas. Puede mantener el estado de cada una de las llamadas y usa protocolos apropiados para enviarle órdenes a los Gateways y a los terminales. Es el punto de control utilizado por la plataforma de aplicaciones para controlar llamadas y aprovechar las capacidades de servicio de la red.

SS7 (Signaling System Number 7): son un conjunto de protocolos de señalización en redes telefónicas conmutadas. Aparte de ofrecer funciones para establecer y terminar llamadas puede ofrecer servicios como traducción de números, mecanismos de cobro prepago y mensajes de texto cortos.

UAM (User Access Module): es el punto de acceso de los usuarios. Es un equipo de la capa de acceso y es encargado de traducir los protocolos de las redes subyacentes en mensajes SIP. También es conocido como Gateway de Señalización.

UMTS (Universal Mobile Telecommunications System): es una tecnología de telefonía celular de tercera generación. Está estandarizado por el 3GPP y utiliza W-CDMA como interfaz de aire.

WSDL (Web Service Description Language): es un formato XML que se utiliza para describir servicios Web. Describe la interfaz pública a los servicios Web. Describe la forma de comunicación, requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios.

RESUMEN

Este trabajo presenta el proceso de desarrollo de un servicio para una red de nueva generación (NGN) utilizando el API de Osa/Parlay. El servicio desarrollado corresponde a un servicio de portabilidad numérica a través de un número virtual y números físicos asociados a él. El servicio se implementó sobre la NGN de EMCALI (red multiservicios). Se desarrolló utilizando la metodología RUP porque era necesario evaluar periódicamente el proceso de desarrollo y el estado actual de la plataforma de servicios de la red multiservicios de EMCALI. El desarrollo del servicio y los resultados obtenidos aportaron pautas al desarrollo del “Modelo arquitectónico para la implementación de nuevos servicios telemáticos sobre la red multi-servicios” presentado por el grupo de investigación GITI de la Universidad Autónoma de Occidente.

INTRODUCCIÓN

Tradicionalmente los servicios implementados en una red de telecomunicaciones han estado estrechamente vinculados con los mecanismos de transmisión y control de datos definidos por las capas inferiores del modelo OSI. Esta característica de las redes tradicionales dificultaba la creación de nuevos servicios sobre las redes ya existentes porque no contaban con suficiente flexibilidad. Esto se refleja en que existen múltiples redes paralelas (IP, PSTN, GSM) o poco relacionadas donde cada una ofrece un tipo de servicio distinto y la interconexión entre ellas resulta difícil y costosa. Estos modelos tradicionales se han visto desplazados en los últimos años con la creciente popularidad de las redes IP y especialmente con la aparición de las redes convergentes o redes de nueva generación.

La ITU-T define las redes de nueva generación como: “Redes basadas en paquetes capaces de proveer servicios de comunicaciones y capaces de usar múltiples anchos de banda, tecnologías de transporte con QoS donde las funciones relacionadas con los servicios son independientes de la capa de transporte bajo ella. Soporta movilidad generalizada que permite una provisión de servicios de servicios a los usuarios de forma oblicua y consistente”.¹

Por esta razón las redes de nueva generación abren un campo de posibilidades a los operadores de redes y empresas prestadoras de servicios. Al separar las funciones relacionadas con servicios de las capas de transporte, se disminuye el tiempo de creación de nuevos servicios porque sus diseñadores no necesariamente deben tener amplios conocimientos en telecomunicaciones o en las tecnologías de la red existente. Esto permite que terceras partes diseñen nuevos servicios sobre las redes de nueva generación otorgando valor agregado a las redes actuales anexadas a ellas.

Actualmente EMCALI cuenta con una red de nueva generación, llamada red multiservicios. Esta red posibilita el acceso a distintos anchos de banda, voz sobre IP y servicios que incluyen: control de llamadas, mensajería y servicios multimedia.

Para aprovechar los recursos entregados por esta red y agilizar el proceso de generación de nuevos servicios el grupo de investigación en telemática de la universidad GITI de la Universidad Autónoma de Occidente propone un Modelo arquitectónico para la implementación de nuevos servicios telemáticos sobre la red multi-servicios.

¹ BERRUTO, Ermanno. Evolución hacia la NGN. Milan, Giugno 13 de 2007. ANFoV – Observatorio NGN. [en línea]

Disponible en Internet:

http://www.anfov.it/s_leNostreAttivita/docs_osservatori/NGN_presentazioneWind_13_6_07.pdf

El presente proyecto presenta el diseño e implementación de un servicio telemático para la red de EMCALI, este proceso aportó pautas para el diseño del modelo arquitectónico mencionado anteriormente. Además, los resultados prácticos obtenidos después de la implementación del servicio fueron usados para la validación de ese modelo. Igualmente con la implementación de este servicio quedó demostrada la capacidad actual de la red multiservicios de EMCALI para desarrollar nuevos servicios por terceras partes.

El servicio fue diseñado utilizando el API de OSA/Parlay. Este API fue desarrollado para crear servicios telemáticos en redes NGN y se caracteriza por implementar un fuerte esquema de seguridad en la que una entidad, llamada Framework, se encarga de autorizar a las aplicaciones el uso de las capacidades de la red subyacente.

La metodología utilizada para el desarrollo del servicio fue la metodología de Proceso Racional Unificado (RUP). Lo que permitió una evaluación periódica del servicio en desarrollo y generó retroalimentaciones tanto al modelo arquitectónico, como a EMCALI acerca del estado de su red multiservicios, sus capacidades y la documentación existente acerca de ella.

Este es uno de los primeros servicios implementados para una red convergente utilizando el API de OSA/Parlay en Latinoamérica, con este desarrollo se abren nuevas posibilidades orientadas al desarrollo de nuevos servicios y el refinamiento de técnicas para el diseño de aplicaciones de este tipo.

En el primer capítulo se presenta la descripción general del proyecto y se cubren los aspectos relacionados con sus participantes, objetivos, marco teórico, justificación y antecedentes. Los modelos de redes de nueva generación o redes convergentes son cubiertos en el segundo capítulo. El tercer capítulo cubre las herramientas disponibles más importantes para el desarrollo de servicios sobre las redes convergentes. El API de OSA/Parlay es analizado en el cuarto capítulo. Los capítulos 5 y 6 cubren el diseño e implementación del servicio telemático. El séptimo capítulo resume las conclusiones aportadas al modelo de desarrollo propuesto por el GITI.

1. DESCRIPCIÓN GENERAL DEL PROYECTO

1.1 TÍTULO DEL PROYECTO

Implementación de un servicio telemático sobre la red multiservicios de EMCALI.

1.2 PARTICIPANTES

1.2.1 Director del Proyecto. Oscar Hernán Mondragón. Ingeniero Electrónico
Master in Wireless and Related Systems.

1.2.2 Participantes. Andrés Felipe Murillo Piedrahita

1.3 DEFINICIÓN DEL PROBLEMA

El modelo que propone NGN de separar la prestación de servicios de las características de la red existente, así como el estímulo para que los servicios sean desarrollados por terceras personas apunta al objetivo principal de ofrecer servicios adicionales, de una forma rápida y económica, para otorgar valor agregado a las prestaciones ya existentes sobre las redes actuales. Esto abre enormes oportunidades para los operadores de telecomunicaciones, habilitando la renovación en la oferta de servicios.²

El modelo arquitectónico propuesto por GITI de la Universidad Autónoma de Occidente agiliza ese proceso de ofrecimiento de nuevos servicios sobre la NGN. Este proyecto implementó un servicio sobre la red multi-servicios para validar el modelo mencionado anteriormente. A través de los resultados prácticos obtenidos durante la fase de implementación se contribuirá con observaciones que aporten al modelo arquitectónico propuesto. Los resultados obtenidos después de la implementación del servicio servirán para validar el modelo arquitectónico propuesto. Este proyecto pretende diseñar e implementar un servicio real sobre la red multiservicios de EMCALI.

El servicio fue diseñado siguiendo la metodología RUP y utilizó como base el análisis realizado por Diana Guevara en “Ambientes de desarrollo para la implementación de nuevos servicios telemáticos sobre la red multi-servicios de EMCALI”, pasantía de la Universidad Autónoma de Occidente CITACION. Los resultados de las iteraciones fueron retroalimentaciones para el modelo

2 FALCARIN, Paolo, LICCIARDI Carlos Alberto y ANDREETTO Alessandra. Service Opportunities for Next Generation Networks [en línea]

Disponible en internet:

http://phoenix.labri.fr/documentation/sip/Documentation/Papers/Programming_SIP/Paper_Publication_and_Draft/V02N01Art263.pdf

arquitectónico propuesto por el. El resultado final validó el modelo propuesto.

1.4 OBJETIVOS

1.4.1 Objetivo General. Diseñar e implementar un servicio que permita validar el Modelo arquitectónico para la implementación de nuevos servicios telemáticos sobre la red multiservicios de EMCALI.

1.4.2 Objetivos específicos.

- Validar los resultados del proyecto “Ambiente de desarrollo para la implementación de nuevos servicios telemáticos sobre la red multiservicios de EMCALI” a través de la implementación de un primer servicio de prueba sencillo.
- Presentar recomendaciones para el modelo de creación de servicios sobre la NGN de EMCALI.
- Diseñar un servicio definitivo basado en el modelo de creación de servicios establecido.
- Implementar el servicio sobre la plataforma NGN de EMCALI.
- Realizar y ejecutar un plan de pruebas del servicio implementado que permita validar el modelo propuesto.

1.5 MARCO TEÓRICO

1.5.1 Evolución de las redes de Nueva Generación

• **Modelo vertical tradicional.** Tradicionalmente los servicios prestados por determinado operador de red o prestador de servicios a un grupo de usuarios estaban directamente vinculados con las tecnologías y protocolos que soportaban la red. Esto implica que para ofrecer servicios adicionales sobre determinada red se debían modificar o ajustar la forma de transportar los datos.

Ese fue el caso de las redes que fueron concebidas originalmente para el transporte de voz únicamente y que con el paso del tiempo, sufrieron algunas modificaciones para habilitar el transporte de datos. Un ejemplo de este suceso es la adaptación que algunas empresas hicieron de sus redes PSTN a ISDN.

Este modelo vertical hacía más costoso y lento el desarrollo de nuevos servicios sobre las redes existentes. Adicionalmente limitaba el aporte que terceros podrían realizar, porque las nuevas aplicaciones debían adaptarse a cada protocolo y a la arquitectura de red en la que funcionaría la aplicación.

Para facilitar el desarrollo de nuevas aplicaciones e integrar las redes de

paquetes IP y las redes tradicionales fue proyectado un modelo de red convergente.

Se ha pasado de un modelo vertical, en el cual la red y los servicios aparecen estrechamente ligados, a un modelo horizontal en el que se propone una independencia absoluta entre ambos y una única solución de red común a todos ellos.

El proceso de evolución en las redes tradicionales hacia las redes de nueva generación se ha dado fundamentalmente motivado por una serie de factores como son:

- La necesidad de reducir los costos respecto a los modelos tradicionales.
 - La necesidad de compartir infraestructuras entre distintas unidades de negocio,
 - La preponderancia cada vez mayor del modelo Internet
 - La necesidad de establecer la convergencia y compatibilidad entre las distintas redes.
 - La necesidad de acelerar el proceso de creación y puesta en funcionamiento de las aplicaciones y servicios.
 - La necesidad de simplificar y unificar la gestión, la operación y el mantenimiento de los servicios.
- **Modelo propuesto por las redes de nueva generación.** Con estas motivaciones las plataformas NGN deben garantizar una arquitectura que permita el desarrollo pertinente y ágil de nuevos servicios sobre las redes existentes. Para lograr esta función las NGN cuentan con una arquitectura, basada en una red IP, que tiene como objetivo el hacer independientes los servicios prestados por determinada red, de la forma como esta transporta los datos. Es decir, cuenta con capas independientes con funcionalidades distintas y que se comunican a través de interfaces.

La función de esas interfaces es abstraer las funciones de las redes subyacentes para ser aprovechadas por la capacidad del servicio. Pero sin que esta capacidad de servicio se encuentre limitada a una tecnología puntual.

1.5.2 Plataformas NGN La ITU-T. Define una red de nueva generación como: “Una red basada en paquetes capaz de proveer servicios de comunicaciones y posibilitar el uso de múltiples anchos de banda, tecnologías de transporte con QoS. Donde las funciones relacionadas con los servicios sean independientes de las tecnologías relacionadas con transporte debajo de

esas funciones. Permite acceso sin trabas a los usuarios hacia las redes y a los proveedores de servicio competentes de su elección. Soporta movilidad generalizada que permite una provisión de servicios consistente y ubicua a los usuarios”³

Esta definición brinda a terceras personas y operadores de red herramientas para renovar los servicios ofrecidos sobre las redes actuales y la posibilidad para ofertar nuevos servicios. Esto amplía las opciones del mercado en el campo de las telecomunicaciones y explota los avances tecnológicos recientes de las redes basadas sobre IP. Las características de una red NGN son:

- Transporte basado en paquetes
- Separación entre la provisión del servicio y el transporte
- Provisión de interfaces abiertas
- Capacidades de banda ancha calidad del servicio de extremo a extremo
- Soporte d servicios de diferentes naturaleza: real time, non real time, streaming, servicios multimedia
- Interoperabilidad e interconexión con redes heredadas a través de interfaces abiertas.
- Movilidad generalizada

1.5.3 Arquitectura de las NGN. La figura 1 muestra la arquitectura básica de una NGN. La topología de estas redes es similar a la de las redes jerárquicas donde existen capas de acceso, distribución y núcleo. Sin embargo se diferencia de la topología tradicional en que la capa de acceso cuenta con dispositivos para hacer la traducción de los protocolos usados por las redes inferiores a paquetes IP y que se usan mensajes SIP para que los elementos de la red, a partir de la capa de acceso, se comuniquen.

De esta forma las capas superiores son independientes de las inferiores, pero conservan el control de los elementos de la red y las llamadas establecidas.

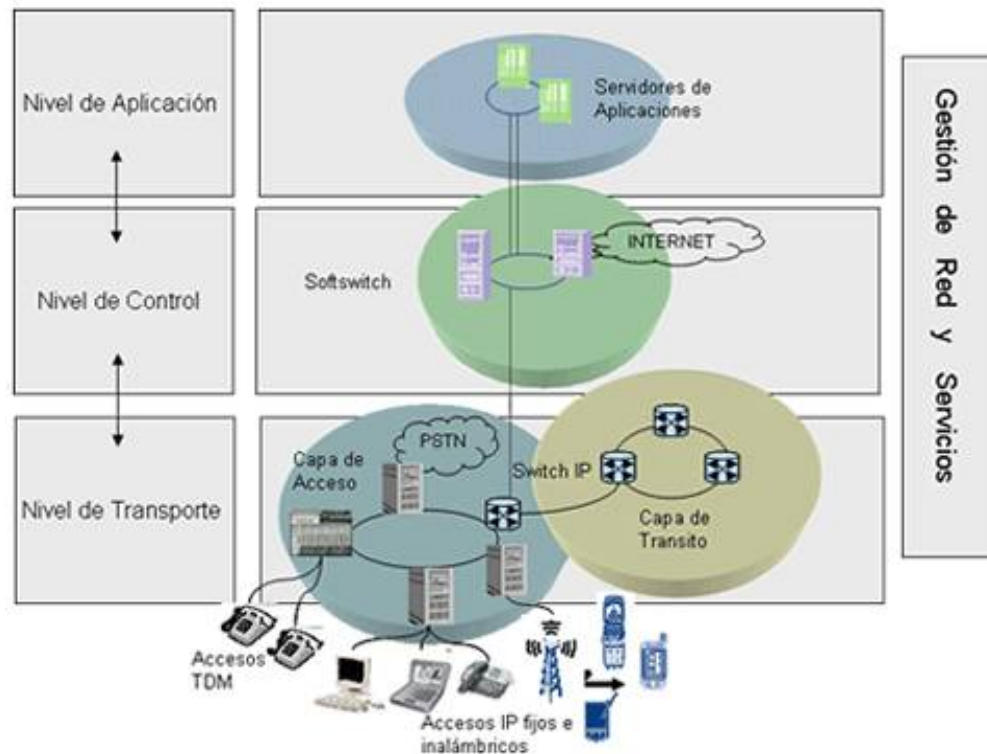
• **Capa de transporte.** Este nivel se divide en dos subniveles: la capa de acceso y la capa de tránsito

- Capa de acceso. Esta capa es la responsable de la conectividad entre los dispositivos finales y el resto de la red. Esta capa utiliza gateways para comunicarse con las redes tradicionales. Adicionalmente cuenta con

3 Recomendación ITU-T 2001. [en línea] Disponible en internet: <http://www.itu.int/ITU-T/ngn/definition.html>

mecanismos para la traducción de los mensajes provenientes de las redes tradicionales a paquetes IP, de esta forma se separan las redes tradicionales de las capas superiores de la red convergente.

Figura 1. Arquitectura típica de una NGN



- **Capa de tránsito.** Esta capa es analógica a la capa de distribución y core en el sistema jerárquico está compuesta por los elementos que se encargan de transportar la mayor parte del flujo de la red hacia las capas superiores.
- **Capa de control.** La capa de control es la capa donde se controlan los recursos y las llamadas que se establecen en la red. Esta estructura centralizada es lo que permite que el desarrollo de aplicaciones pueda controlar las llamadas establecidas en la red subyacente. Las funciones de esta capa junto con la capa de acceso son las que posibilitan que las aplicaciones desarrolladas puedan usarlas usuarios de distintas redes.

Como se puede apreciar, este tipo de control de las aplicaciones requiere de un esquema de seguridad que sólo autorice a las aplicaciones indicadas el control de las llamadas.

Esta capa incluye las siguientes funciones:

- Control de transporte. Incluye el control de recursos, admisiones y enlaces a la red.
- Control de recursos y admisión. Permite que los servicios determinados accedan a las funciones de transporte de la red. Esta decisión se basa en información de suscripción, acuerdos a nivel de servicio, políticas de la red, prioridad de servicios y el estado actual de los recursos.
- Funciones de control de acceso a la red. Estas funciones proveen registro para que los usuarios finales puedan acceder a los servicios y aplicaciones de NGN.
- **Capa de aplicaciones.** Esta capa implica las funciones de control de servicios y las aplicaciones, también funciones de soporte a servicios y aplicaciones.
 - Funciones de control de servicios. Estas funciones autorizan los servicios (SCF Service Capability Feature) dentro de la NGN. Se controla el registro de esos servicios, su autenticación y los recursos que administran.
 - Funciones de control de aplicaciones. Estas funciones son las que controlan las aplicaciones que pueden pertenecer a la red NGN, así como buscar y acceder a los servicios autorizados.
 - Funciones de soporte de aplicaciones y servicios. Estas funciones son las que permiten que las aplicaciones registradas utilicen los servicios que han sido autorizados.

1.5.4 Componentes de una plataforma NGN. Una NGN tiene una serie de componentes específicos que implementan las funciones de las capas mencionadas anteriormente. Para el caso de la red multi servicios de EMCALI estos componentes fueron adquiridos a la empresa estatal china ZTE.

- **Softswitch.** Los softswitch son una parte fundamental de las redes NGN. En estos switches se controla el transporte de datos y se realizan las funciones correspondientes a la capa de control y las funciones de control de la capa de aplicación mencionadas anteriormente. Las funciones del softswitch son:
 - Control y señalización de los diferentes tipos de llamadas (voz, video, datos, etc.)
 - Contacta a los servidores de aplicaciones para proveer los servicios solicitados

- Realiza el control de las puertas de enlace y los terminales de usuario
- Interconexión con la RTPC a través de las puertas de enlace de señalización.⁴
- **Puertas de enlace.** Las puertas de enlace son las que permiten la conectividad de las redes NGN con otras redes. Las puertas de enlace sirven como interfaces entre la red núcleo de NGN. Sus funciones son:
 - Permiten la interconexión de la NGN con el mundo de conmutación de circuitos de la RTPC
 - Las Gateways Troncales conectan la NGN con la RTPC a través de enlaces troncales
 - Las Gateways de Señalización permiten la comunicación del softswitch con la red de SS7 de la
 - RTPC u otras redes mediante la traducción de la señalización
 - Las Gateways de Acceso permiten la conexión a la red de líneas tradicionales
 - Las Gateways Residenciales se ubican del lado del usuario y permiten la conexión de teléfonos análogos o digitales a la NGN.⁵
- **Servidores de aplicaciones.** Estos servidores son los que proveen la lógica de las aplicaciones. Estos servidores responden a las solicitudes de aplicaciones que realizan los usuarios. Una característica del modelo propuesto por las NGN es que estos servidores no necesariamente deben ser propietarios de la empresa proveedora de comunicaciones. Estos servidores pueden ser propietarios de terceras personas y luego de firmar un acuerdo de servicio acceder a la red NGN. Las aplicaciones de esos servidores también serán controladas por la capa de aplicaciones mencionada anteriormente.

1.5.5 Red multi-servicios de EMCALI. EMCALI posee una red convergente de nueva generación, llamada red Multi-servicios. Con esta red se proveen accesos de banda ancha, servicios LAN to LAN, servicios de conexión a Internet, servicios de voz a través de líneas POTS, voz sobre IP, telefonía sobre IP y demás servicios como: servicios de red inteligente, correo de voz, mensajería unificada, servicios prepago, servicios de IVR y televisión sobre IP. Esta red multi-servicios cuenta con servidores de aplicaciones que soportan la arquitectura OSA/Parlay para la creación de nuevos servicios y aplicaciones.

4 Departamento de Planeación e Ingeniería EMCALI. Marco Teórico RMS EMCALI 1. Santiago de Cali. 2007 p4

5 Ibid p5

1.5.6 Herramientas para el desarrollo de servicios sobre NGN. Uno de los objetivos de NGN es el desarrollo de nuevos servicios y aplicaciones por terceros. Se ha favorecido el uso de interfaces abiertas y de software libre para el desarrollo de aplicaciones con el objetivo de agilizar el proceso de creación y renovación de servicios, en un entorno competitivo que les brinde valor agregado a las redes existentes.

- **API.** Una variedad y tipos de APIS están surgiendo en los productos NGN, proveyendo diferentes niveles de abstracción funcional.⁶

Algunos sistemas proveen varios niveles de abstracción de API, la diferencia entre un nivel de abstracción y otro es que entre más baja sea la abstracción se tiene más control sobre los detalles específicos de la red y los protocolos, pero se incrementa la complejidad de los diseños.

- OSA/Parlay. Son muchas las herramientas disponibles para el desarrollo de servicios para NGN. Una de estas alternativas que poco a poco gana popularidad entre los diseñadores de servicios sobre estas redes es el uso del API de OSA/Parlay, cuenta con las ventajas de las API mencionadas anteriormente y además tiene la característica de ser una interfaz estándar abierta y que cuenta con un equipo enorme de desarrolladores que están detrás del continuo mejoramiento de la API. Este grupo de desarrolladores es el grupo Parlay.
- API de Parlay. OSA/Parlay define una arquitectura que habilita el trabajo interno entre aplicaciones IT y las características de las telecomunicaciones en redes móviles a través de una interfaz abierta estandarizada. La funcionalidad de la red es descrita como Funciones de Capacidad de Servicio (SCF) y las aplicaciones pueden ser desarrolladas por un dominio administrativo de tercera persona. SCF proveen acceso a las capacidades de la red que un operador de red necesita exportar a través de una interfaz OSA.

Las ventajas del uso de APIs de OSA/Parlay son:

- Esconde la complejidad de la red, los protocolos y su implementación específica.
- Es apropiado para el desarrollo de aplicaciones por terceras personas, se requiere cierto conocimiento en telecomunicaciones.
- Provee un acceso seguro y controlado a las capacidades de la red

⁶ FALCARIN Paolo y LICCIARDI Carlos Alberto. Análisis de tecnologías para el desarrollo de servicios en NGN. [en línea]

Disponible en Internet <http://softeng.polito.it/falcarin/docs/Annals03.pdf>

provistas por un proveedor de red a terceras personas proveedoras de servicios.

- Expone casi todas las capacidades de la red provistas por el protocolo de comunicaciones correspondiente y facilita el desarrollo de servicios combinando varias capacidades de servicio e integrando aplicaciones IT.
- La madurez de OSA/Parlay se ha vuelto significativa, principalmente, después de la creación de un grupo de trabajo conjunto que incluye 3GPP, ETSI, Parlay y JAIN.⁷
- Capacidades de servicio en OSA/Parlay. Estas capacidades de servicio, son las funciones principales de la API de OSA/PArlay, la versión actual de esta API es la 6.
- Autenticación de aplicaciones en OSA/Parlay. Para que las aplicaciones y las capacidades de servicios puedan comunicarse, debe realizarse un proceso de autenticación con el Framework, esto es lo que ofrece un entorno seguro para las aplicaciones. Para esto, las capacidades de servicio se registran en el Framework. Después cuando una aplicación requiere de determinada funcionalidad de la red, se conecta con el Framework, se autentica y solicita acceso a la capacidad del servicio. El Framework después de autenticar y autorizar a la aplicación genera una instancia a la capacidad de servicio solicitada y le proporciona la interfaz a la aplicación.
- **Parlay X.** Parlay X es una forma simplificada de Parlay para que pueda ser usado por desarrolladores que sean necesariamente profesionales del conocimiento interno de las redes de comunicaciones.

Parlay X ofrece una segunda capa de abstracción que facilita la interacción para la aplicación con la red. Sigue siendo una puerta de enlace, la interfaz está basada en servicios Web se comunica directamente con la red subyacente. Parlay X realiza de forma general todas las tareas necesarias de autenticación y autorización respecto al framework, de esta forma las aplicaciones se centran más en la lógica del servicio más que en el acceso a la red.

- **Parlay Client Hub (PCH).** Es una tecnología propietaria de la empresa estatal China ZTE. PCH provee un nivel de abstracción superior a Parlay. De esta forma se facilita el desarrollo de aplicaciones porque muchas funciones en Parlay como el registro con el Framework son realizadas por el PCH.

El objetivo de PCH es unificar el procesamiento de interfaces entre las

⁷ Ibid

aplicaciones y el Framework, así como entre aplicaciones y SCF's para que los desarrolladores se enfoquen solamente en la lógica de la aplicación.⁸

- **Lenguajes de script.** Estos lenguajes son livianos, altamente personalizables y típicamente interpretados, apropiados para el área de desarrollo rápido de aplicaciones. Brindan conexiones entre los componentes existentes. Estas características les permiten ser usados para codificar o cambiar aplicaciones durante el tiempo de ejecución, e interactuar con programas en funcionamiento.⁹

- **SCE.** Usar entornos de creación de servicios permite un desarrollo rápido de servicios con pocos conocimientos acerca de los protocolos de red; la lógica de los servicios es diseñada usando diagramas de flujo de componentes básicos ofrecidos por el SCE. Estos componentes ofrecen conectividad con pilas de diferentes protocolos, enlazándolos a código externo y a bases de datos.

Las funciones de PCH son:

- **Manejo de sesiones.** PCH administra las sesiones de todas las aplicaciones ante el Framework. Es a través de estas sesiones que se puede utilizar la funcionalidad de las capacidades de servicio por parte de las aplicaciones, estas capacidades de servicio incluyen aquellas que usa el Framework para autenticar la aplicación, descubrir y utilizar otras capacidades de servicio.
- **Administración de aplicaciones.** PCH provee servicios comunes para aplicaciones cliente. Es a través de estas aplicaciones que se realiza la configuración de las aplicaciones.
- **Manejo de hilos.** Para incrementar la eficiencia de la aplicación. PCH cuenta con un "pool" de hilos para procesar los llamados a capacidades de servicio. Las características de este pool pueden ser configuradas.
- **Conectividad con la base de datos.** PCH ofrece herramientas para que las aplicaciones se comuniquen con la base de datos de la plataforma de aplicaciones de forma segura y eficiente. En el caso de EMCALI la base de datos instalada es Oracle 9i

1.6 ANTECEDENTES

1.6.1 PLATAFORMA MULTISERVICIOS DE EMCALI. Las Empresas Municipales de Cali (EMCALI) como proceso de innovación en

⁸ GUEVARA, Diana Marcela. Ambientes de desarrollo para la implementación de nuevos servicios telemáticos sobre la red multi-servicios de EMCALI. Pasantía para Ingeniera Electrónica Santiago de Cali: Universidad Autónoma de Occidente. Facultad de Ingeniería. 2008. p 82

⁹ Ibid

telecomunicaciones e integrándose a operadores de telecomunicaciones de vanguardia en el tema de NGN, implementó su plataforma multi-servicios con la participación de la empresa ZTE. Este proceso integró los proyectos de sustitución de 95.230 líneas entre analógicas y digitales por la tecnología IP y la ampliación de líneas de telefonía básica para ofertar 44800 líneas nuevas.¹⁰

La empresa estatal china ZTE Corporation, ganó la licitación para el montaje de la red multi-servicios de telecomunicaciones de EMCALI. Esta red de alta tecnología posibilita agregar servicios de voz, datos e imagen con nitidez, velocidad y capacidad inmediata de transmisión.

Los usuarios podrán acceder a canales de comunicación interactiva como teletrabajo, sistemas de seguridad, medios de entretenimiento, comunicaciones sobre protocolo de Internet, locales, nacionales e internacionales, teleconferencias, video-conferencias, video por demanda y toda una serie de opciones en desarrollo por parte de los proveedores de contenido, contando con redes de banda ancha e Internet de altísima velocidad con más de dos megas de capacidad

1.6.2 Servicios implementados previamente usando OSA/Parlay.

Anteriormente en la universidad se han desarrollado proyectos que implementan servicios en torno a NGN. En el año 2007, dos estudiantes realizaron el diseño de un servicio telemático para la red NGN usando OSA/Parlay, este consiste en una línea de opinión de usuarios acerca de diversos temas de interés.

La implementación de este servicio sobre la plataforma multi-servicios de EMCALI no se realizó, ya que no se contaba con el conocimiento necesario para hacerlo, el servicio fue probado sobre la plataforma Ericsson NGR, que también soporta OSA/Parlay, sin embargo nunca se realizaron pruebas reales sobre la red multiservicios¹¹

1.6.3 Modelo arquitectónico para la implementación de nuevos servicios telemáticos sobre la red multi-servicios de EMCALI. El grupo de investigación en Telemática de la Universidad Autónoma (GITI de la Universidad Autónoma de Occidente) actualmente está desarrollando un proyecto de investigación que pretende crear un modelo arquitectónico para la implementación de servicios telemáticos sobre la red multiservicios de EMCALI.

Los realizadores de este proyecto son el ingeniero Oscar Mondragón y la ingeniera Zeida Solarte.

Este proyecto surge por la necesidad de conocer y dominar en alto grado la

¹⁰ Ibid. p38

¹¹ VALLECILLA, Víctor Manuel y CARMONA, Marcela. Diseño e implementación de servicios telemáticos sobre la red inteligente NGN de EMCALI. Pasantía para Ingeniería informática .Santiago de Cali: Universidad Autónoma de Occidente. Facultad de ingeniería. Departamento de sistemas de información. 2007. 256p

arquitectura propuesta por OSA/Parlay y establecer claramente la forma de implementarla sobre la plataforma ZTE. El modelo arquitectónico le permitirá a EMCALI un desarrollo rápido de servicios telemáticos que puedan ser fácilmente desplegados sobre la infraestructura de red.

1.6.4 Ambientes de desarrollo para nuevos servicios telemáticos sobre la red multi-servicios de EMCALI. Actualmente está culminando un proyecto que realiza un análisis sobre los ambientes de desarrollo disponibles para la red multiservicios de EMCALI. Del análisis realizado se seleccionará un ambiente de desarrollo y se especificarán las herramientas y características de este ambiente.

Para esto hubo una fase inicial donde se realizó la captura de requerimientos para la construcción de nuevos servicios telemáticos en la red multiservicios de EMCALI. Posteriormente se analizó y documentó la arquitectura NGN actual de EMCALI. Después se analizaron las diferentes tecnologías para la creación de servicios de nueva generación y se eligió como la más adecuada para la arquitectura el API de OSA/Parlay y el Parlay Client Hub (PCH).

1.6.5 Otras implementaciones de NGN. En el Reino Unido, BT transfirió en noviembre de 2006 las primeras líneas de abonado a su red NGN bautizada 21st Century Network (Red del siglo XXI). A finales del verano de 2007 unos 350 000 hogares de la zona también estarán conectados. Ese proyecto servirá como evaluación antes de lanzar un proyecto para que todos los clientes se encuentren conectados a la NGN, es decir unos 30 millones de líneas abastecidas por más de 5.500 centrales telefónicas.¹²

Net4Call, una empresa de Noruega que desarrolla soluciones de comunicaciones para redes fijas, móviles y operadores de red IP; desarrolló un PBX virtual con operación nacional. Usando la arquitectura de Net4Call, el operador fue capaz de desarrollar el servicio en un tiempo muy corto utilizando tecnología estándar J2EEE. El PBX virtual es completamente personalizable, facilita y satisface las necesidades de comunicaciones de pequeñas, medianas y grandes empresas.¹³

Este PBX fue desarrollado en software y hardware estándar que puede ser conectado a cualquier puerta de enlace de Parlay/OSA o a un servidor SIP.

12 COHEN, Tracy y HERNEDEZ, Janet [en línea]. El camino hacia las redes de próxima generación-pioneros [en línea]. Disponible en Internet:
<http://www.itu.int/itunews/manager/display.asp?lang=es&year=2007&issue=02&ipage=next-generation2&ext=html#top>

13 Net4Call Press Release. Octubre de 2005. [en línea] Disponible en Internet:
http://www.parlay.org/imwp/idms/popups/pop_download.asp?contentID=7019

1.7 JUSTIFICACIÓN

Los proveedores de servicios de telecomunicaciones han encontrado que es necesario investigar profundamente en tecnologías IP que implementen NGN para posicionarse a sí mismos y ser capaces de ofrecer nuevos servicios. Responder a las demandas más fuertes por innovación. Los proveedores de servicios necesitan ofrecer nuevos, y diferenciados servicios para permanecer siendo competitivos.¹⁴

Teniendo en cuenta lo anterior, resulta necesario para mantener la competitividad el invertir en tecnologías diseñadas para NGN. Sin embargo, junto con la inversión realizada debe haber un interés por entender su tecnología, las herramientas que brinda y encontrar modelos que permitan el rápido desarrollo de nuevos servicios sobre la misma. Estos esfuerzos son los que permiten aprovechar las capacidades de las NGN, porque le otorgan valor agregado a las redes existentes a través de la prestación de nuevos servicios. Los servicios son parte vital de cualquier red y un área crítica de diferenciación¹⁵

Esta tendencia a adoptar la creación y renovación de servicios como política de crecimiento de las organizaciones ha ganado popularidad en las empresas proveedoras de telecomunicaciones, así como el desarrollo de arquitecturas para desarrollar esos servicios. Más y más proveedores de servicios de telecomunicaciones están transformando su forma de creación de servicios y desarrollando sus propios servicios a la vez que se relacionan con terceras personas. El resto, entonces, es desarrollar arquitecturas que soportan una forma de desarrollo rápida, económica y orientada al servicio para NGN.¹⁶

EMCALI adquirió a la empresa ZTE la plataforma multiservicios, esto representó una gran inversión para la empresa. Sin embargo, las capacidades de una plataforma NGN son aprovechadas efectivamente cuando se proveen nuevos servicios, no sólo cuando se usa su infraestructura para disminuir costos, que es el uso actual de la plataforma. Para brindar un aprovechamiento efectivo de esta plataforma se realizó un proceso de evaluación de la plataforma y posterior desarrollo de aplicaciones sobre ella, el trabajo de este proyecto corresponde a la primera de esas aplicaciones.

Para acelerar el proceso de creación de servicios sobre la NGN de EMCALI, el GITI de la Universidad Autónoma de Occidente desarrolló un modelo arquitectónico para la implementación de nuevos servicios telemáticos. Como

14 REHEMTULLA, Mebs y HUGHES Glenn. Service creation: Meeting the product lifecycle challenge. Enero de 2006. IBM Product Lifecycle management. [en línea]
Disponible en internet:

http://www03.ibm.com/industries/telecom/doc/content/bin/Service_Creation_Meeting_the_product_lifecycle_challenge.pdf

15 UNMEHOPA Musa, VEMURI Kumar, BENNET Andy. Parlay/OSA: From Standards to Reality. Wiley 2006. p 31

16 Ibid

apoyo al desarrollo de este modelo, se diseñó de forma simultánea un servicio real implementado sobre la NGN.

Este servicio se desarrolló utilizando el método iterativo RUP. Este método al estar determinado por un número de fases, permite al final de cada una de ellas entregar un avance concreto del trabajo realizado. Los conocimientos adquiridos y las recomendaciones realizadas a medida que se desarrolla el modelo arquitectónico alimentaron el proceso de desarrollo del servicio. Igualmente, las observaciones encontradas durante el diseño e implementación del mismo, fruto de cada entrega parcial por cada fase del RUP; sirvieron como retroalimentación para el modelo propuesto, ya que este servicio se desarrollará en condiciones reales.

De esta forma el modelo planteado no es modelo teórico. También estuvo retroalimentado por un servicio implementado en un entorno real, lo que valida el modelo propuesto por el GITI de la Universidad Autónoma de Occidente .

2. REDES DE NUEVA GENERACIÓN (NGN)

Las redes de nueva generación ó redes convergentes surgen como una alternativa, motivada por la creciente popularidad del modelo de Internet, para agilizar el desarrollo de nuevos servicios sobre las redes de telecomunicaciones y para integrar el tráfico de los distintos tipos de redes existentes. En este capítulo se analiza la arquitectura propuesta por este tipo de redes, se realiza una reseña acerca de su historia y se explica como se encuentra configurada la red multiservicios de EMCALI que es una red NGN.

2.1 HISTORIA DE LAS REDES DE NUEVA GENERACIÓN

2.1.1 Redes tradicionales. Las primeras redes de telecomunicaciones que se extendieron por el mundo con fines comerciales fueron las redes de telefonía fija, comúnmente conocidas como PSTN (por sus siglas en inglés Public Switched Telephone Network). Estas redes tradicionales usan señales análogas para llevar la información desde un extremo a otro. Sin embargo, en este proceso de transporte de la información interceden diversos equipos que administran los recursos disponibles de la red. Adicionalmente se establece una ruta para que la información viaje desde un extremo a otro, esta ruta generalmente no es permanente, se mantiene mientras dure la comunicación. Por esa razón es necesario que los equipos que interceden en la comunicación mantengan información actual sobre el estado de las llamadas, esto se conoce como procesamiento de llamada¹⁷. Así desde las primeras redes de comunicación se crea la noción de procesamiento de llamadas, de esta forma el proceso de una llamada puede ser descrito como una máquina de estados.

Con la madurez que adquirieron estas redes, la necesidad de nuevos servicios fue más pronunciada. Sin embargo en las redes tradicionales y las redes de circuitos que soportaban líneas de Internet a través de módems o las primeras versiones de DSL existían notables problemas para el desarrollo de nuevos servicios. Para la adición de nuevos servicios se debían solucionar primero problemas de compatibilidad y extensas pruebas para verificar que los servicios funcionarían de forma adecuada y que los múltiples servicios existentes no interactuaran y se comportaran de una forma impredecible. Para atender y solucionar esos problemas se propuso un nuevo modelo de red en los años 1980, este modelo fue conocido como “Red Inteligente”.

2.1.2 Evolución hacia la red inteligente. Uno de los primeros ejemplos de servicios que llevaron a crear el modelo de Red Inteligente fue un servicio introducido en la década de 1980 en Estados Unidos. Estos fueron los servicios 800¹⁸. Estos números no se conectan directamente a un equipo final en la red sino que requieren ser traducidos a una línea telefónica actual para

17 Op cit Unmehopa. p 5

18 JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 5

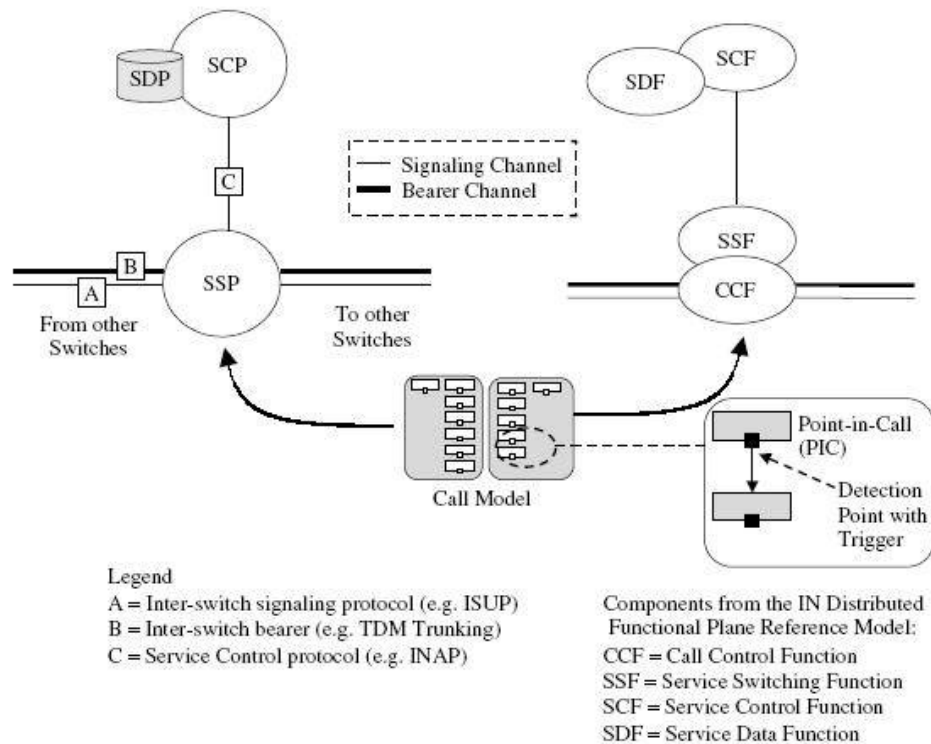
que la llamada sea enrutada. Una característica de este servicio fue que la base de datos de a qué líneas debería traducirse según el número de la serie 800 fue almacenada en una base de datos centralizada. De esta forma no todos los switches tenían que almacenar la base de datos sino que hacían peticiones a través de Canales de Señalización Común (CCS).

Este fue uno de los primeros servicios que propuso un modelo donde existía un elemento de la red dedicado a ejecutar la lógica de una aplicación y que se encontraba separado de los elementos de control de llamadas mencionados anteriormente.

Después de el desarrollo de esa aplicación Bellcore de RBOC y At&T definieron el modelo de Red Inteligente.

2.1.3 Red inteligente. En las redes inteligentes los switches no sólo son dispositivos que ejecutan las máquinas de estado correspondientes a las llamadas que establecen los usuarios. La lógica del servicio es separada de la lógica del control de las llamadas. Los elementos encargados de ejecutar esta lógica son conocidos como Punto de Control de Servicio o SCP (por sus siglas en inglés). De esta manera los switches ahora poseen dos funciones: por un lado una función llamada Función de Control de Llamadas (CCF), esta función ejecuta las funciones básicas de la máquina de estados de la llamada. La segunda función es la Función de Conmutación de Servicio (SSF) que está encargada de la capacidad del servicio de interactuar con el SCP, realizar solicitudes, recibir respuestas y demás. Los switches de servicio también son llamados SSP. La figura 2 muestra este modelo.

Figura 2. Modelo de servicio de Red Inteligente



Fuente: UNMEHOPA Musa, VEMURI Kumar, BENNET Andy. Parlay/OSA: From Standards to Reality. Willey 2006. p 6

A pesar del avance logrado por la arquitectura de AIN estas redes aún presentaban dificultades en la generación de nuevos servicios. Por un lado a pesar de proponer una arquitectura más distribuida, no lograba separar de forma adecuada las capas de control de las llamadas y la capa de aplicaciones. Adicionalmente las herramientas usadas para el desarrollo de nuevos servicios requerían un amplio conocimiento de telecomunicaciones y los protocolos de red de las redes subyacentes lo que limitaba las posibilidades de personal capacitado para generar nuevas aplicaciones.. La integración de las nuevas aplicaciones a sistemas antiguos que ya habían sido implementados era un problema común con este modelo. Cada nueva aplicación necesitaba interactuar con esos sistemas, cada punto de interacción representaba dinero. Adicionalmente las aplicaciones tenían que ser construidas una y otra vez para cada proveedor de servicio, debido a las diferencias entre las redes.

Como se puede apreciar, ese modelo vertical hacía más costoso y lento el desarrollo de nuevos servicios sobre las redes existentes, de igual manera, limitaba el aporte que terceras personas podrían realizar debido a las modificaciones que deberían hacerse a los protocolos y tecnologías debajo de esos servicios.

En resumen, las siguientes fueron las necesidades que impulsaron el desarrollo de una red de nueva generación ó red convergente.

- Reducir los costos respecto a los modelos tradicionales.
- Compartir infraestructuras entre distintas unidades de negocio.
- Adaptar las redes tradicionales al popular modelo de Internet.
- Establecer la convergencia y compatibilidad entre las distintas redes.
- Acelerar el proceso de creación y puesta en funcionamiento de las aplicaciones y servicios.
- Simplificar y unificar la gestión, la operación y el mantenimiento de los servicios.

Con estas motivaciones las plataformas NGN deben garantizar una arquitectura que permita el desarrollo pertinente y ágil de nuevos servicios sobre las redes existentes. Para lograr esta función las NGN cuentan con una arquitectura, basada en una red IP, y diversos Gateways que tienen como objetivo el hacer independientes de la red los servicios prestados.

2.2 PLATAFORMAS NGN.

Con la explosión de Internet y el desarrollo de protocolos estándares y software comercial de bajo costo se abrió la posibilidad de desarrollar redes basadas en paquetes que ofrecieran más ventajas para realizar llamadas que las tradicionales redes conmutadas. “Las Redes Convergentes ó Redes de Nueva Generación son un esfuerzo por tomar ventaja de las redes tradicionales PSTN y potenciar el desarrollo de las redes basadas en paquetes.”¹⁹

La ITU-T define una red de nueva generación como: “Una red basada en paquetes capaz de proveer servicios de comunicaciones y posibilitar el uso de múltiples anchos de banda, tecnologías de transporte con QoS. Donde las funciones relacionadas con los servicios sean independientes de las tecnologías relacionadas con transporte debajo de esas funciones. Permite acceso sin trabas a los usuarios hacia las redes y a los proveedores de servicio competentes de su elección. Soporta movilidad generalizada que permite una provisión de servicios consistente y ubicua a los usuarios”.

Esta definición brinda a terceras personas y operadores de red herramientas para renovar los servicios ofrecidos sobre las redes actuales y la posibilidad para ofertar nuevos servicios. Esto amplía las opciones del mercado en el campo de las telecomunicaciones y explota los avances tecnológicos recientes de las redes basadas sobre IP. Adicionalmente los servicios desarrollados para estas redes pueden ser aprovechados por redes de varios tipos incluyendo redes conmutadas, esto dinamiza el negocio de la prestación de servicios en

¹⁹ Op Cit JAIN p 18.

las telecomunicaciones.

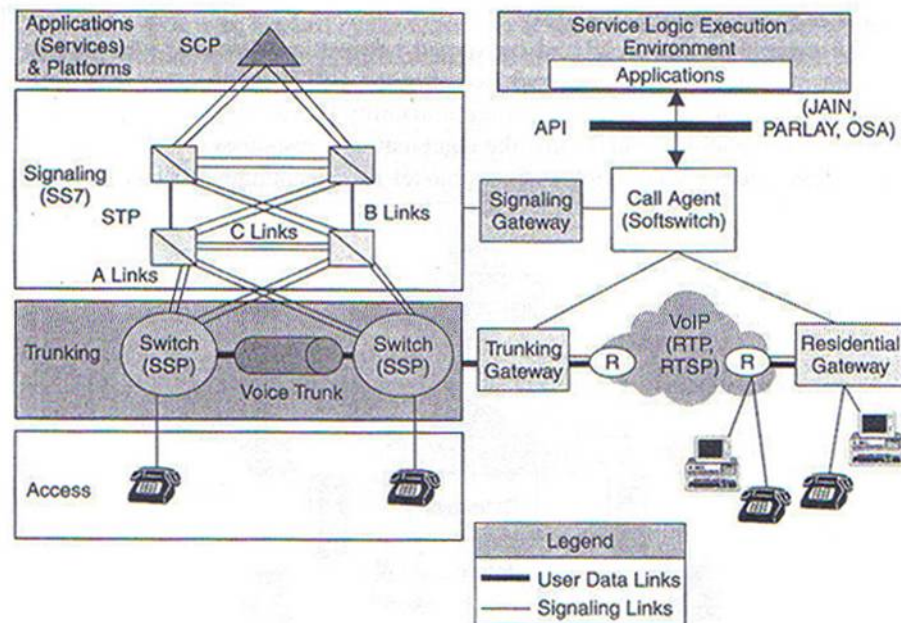
Las características de una red NGN son:

- Transporte basado en paquetes
- Separación entre la provisión del servicio y el transporte
- Provisión de interfaces abiertas
- Capacidades de banda ancha calidad del servicio de extremo a extremo
- Soporte de servicios de diferentes naturaleza: real time, non real time, streaming, servicios multimedia
- Interoperabilidad e interconexión con redes heredadas a través de interfaces abiertas.
- Movilidad generalizada entre las redes tradicionales y las basadas en IP²⁰

La figura 3 muestra un esquema básico de las redes NGN y su relación con la red conmutada de circuitos.

2.2.1 Arquitectura de las NGN. Estas redes cuentan con las siguientes capas. Capa de tránsito. Esta capa es analógica a la capa de distribución y núcleo en el sistema jerárquico está compuesta por los elementos que se encargan de transportar la mayor parte del flujo de la red hacia las capas superiores.

Figura 3. Arquitectura de Red de Nueva Generación o Red Convergente



Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005.p 18

• **Capa de control.** La capa de control es la capa donde se controlan los recursos y las llamadas que se establecen en la red. Esta estructura centralizada es lo que permite que el desarrollo de aplicaciones pueda controlar las llamadas establecidas en la red subyacente. Las funciones de esta capa junto con la capa de acceso son las que posibilitan que las aplicaciones desarrolladas puedan usarlas usuarios de distintas redes.

Como se puede apreciar, este tipo de control de las aplicaciones requiere de un esquema de seguridad que sólo autorice a las aplicaciones indicadas el control de las llamadas.

Esta capa incluye las siguientes funciones:

- Control de transporte. Incluye el control de recursos, admisiones y enlaces a la red.
- Control de recursos y admisión. Permite que los servicios determinados accedan a las funciones de transporte de la red. Esta decisión se basa en información de suscripción, acuerdos a nivel de servicio, políticas de la red, prioridad de servicios y el estado actual de los recursos.
- Funciones de control de acceso a la red. Estas funciones proveen registro para que los usuarios finales puedan acceder a los servicios y aplicaciones de NGN.

- **Capa de aplicaciones.** Esta capa implica las funciones de control de servicios y las aplicaciones, también funciones de soporte a servicios y aplicaciones.

- Funciones de control de servicios. Estas funciones autorizan los servicios (SCF Service Capability Feature) dentro de la NGN. Se controla el registro de esos servicios, su autenticación y los recursos que administran.

- Funciones de control de aplicaciones. Estas funciones son las que controlan las aplicaciones que pueden pertenecer a la red NGN, así como buscar y acceder a los servicios autorizados.

- **Funciones de soporte de aplicaciones y servicios.** Estas funciones son las que permiten que las aplicaciones registradas utilicen los servicios que han sido autorizados.

2.2.2 Componentes de las NGN. Cada una de las capas mencionadas anteriormente cuenta con un equipo característico que cumple las funciones descritas anteriormente.

- Softswitch. Todos los gateways y la funcionalidad de la red IP son controlados por una entidad llamada “Agente de llamada”. En las redes NGN este agente se conoce con el nombre de softswitch, este agente mantiene el estado de cada llamada y utiliza protocolos especiales para enviar comandos apropiados a los gateways. En Osa/Parlay este agente envía notificaciones a la plataforma de aplicaciones cuando hay llamadas que cumplen con el criterio de la notificación. También recibe comandos de la plataforma de aplicaciones.

- Puertas de enlace. Estos elementos permiten la conectividad de la red de núcleo con las redes subyacentes en una NGN, como tal son el punto de acceso de esas redes a la red convergente. Estas puertas de enlace se caracterizan porque realizan la traducción de SIP y otros lenguajes de señalización usados en las capas superiores con los protocolos propios de redes de circuitos, GSM e IP.

- Servidores de aplicaciones. Estos elementos almacenan la lógica de las aplicaciones. A estos servidores los usuarios realizan peticiones cuando se solicita determinada aplicación o servicio. Una característica especial de las redes convergentes es que gracias a su esquema de seguridad es posible que estos servidores se encuentren por fuera de la plataforma NGN propietaria.

2.3 RED MULTISERVICIOS DE EMCALI

EMCALI tiene en servicio una red convergente, de nueva generación, llamada

red Multi-servicios. Con esta red proveen accesos de banda ancha, servicios LAN to LAN, servicios de conexión a Internet, servicios de voz a través de líneas POTS, voz sobre IP, telefonía sobre IP y demás servicios como: servicios de red inteligente, correo de voz, mensajería unificada, servicios prepago, servicios de IVR y prontamente televisión sobre IP. Esta red multi-servicios cuenta con una plataforma de aplicaciones que soporta la arquitectura OSA/Parlay para la creación de nuevos servicios y aplicaciones.

La red multiservicios de EMCALI se encuentra compuesta por cuatro capas, en cada una de ellas se encuentran dispositivos que cumplen funciones específicas.

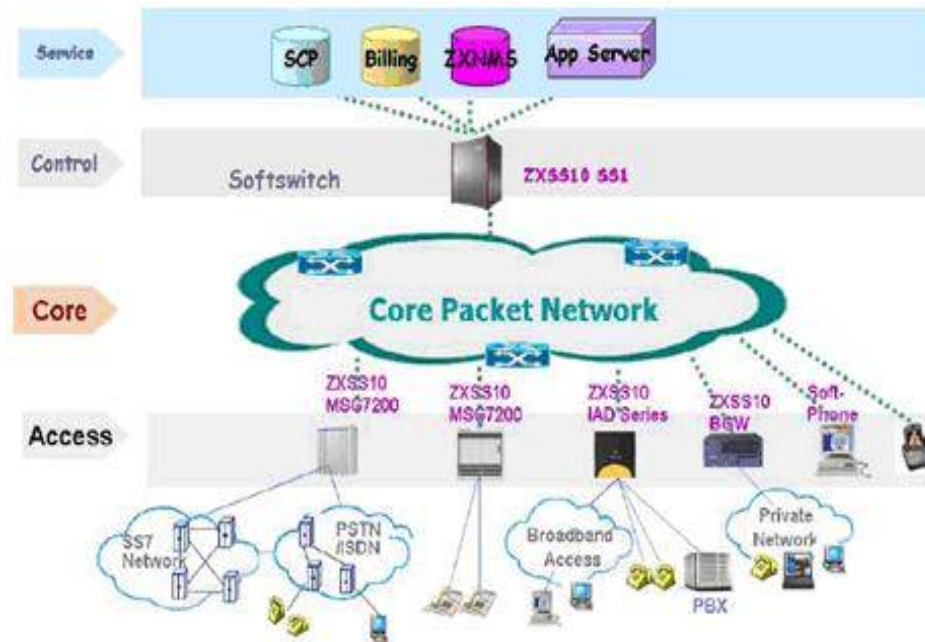
2.3.1 Componentes de la red multi-servicios de EMCALI. La red multi-servicios está conformada en capas, como se indica en la figura 4.

- **Capa de acceso.** La capa de acceso está compuesta por:

Unidades de Acceso Multi-servicios (UAM) que pueden brindar servicios de: telefonía convencional, RDSI, Accesos de banda ancha XDSL y accesos Ethernet. Estas UAM se conectan a la red de acceso.

La red PSTN existente conformada por centrales telefónicas. La red PSTN se integra a la red convergente a través de los Gateway troncales y de señalización conectados a dos centrales tándem.

Figura 4. Red Multiservicios de EMCALI



Fuente: GUEVARA, Diana Marcela. Ambientes de desarrollo para la implementación de nuevos servicios telemáticos sobre la red multi-servicios de EMCALI. Pasantía para Ingeniera Electrónica Santiago de Cali: Universidad Autónoma de Occidente. Facultad de Ingeniería. 2008. p 32

La red GSM está constituida por radio bases (BTS), una controladora de radio bases (BSC) y un switch de conmutación móvil (MSC). La red GSM se integra a la red convergente a través de Gateways troncales y se conecta a la red PSTN a través de centrales tándem.

Red de acceso, conformada por anillos de switches de nivel 3 enlazados por fibra óptica con capacidad de 10 Gbps cada uno. La red de acceso conecta las UAM, clientes con acceso Ethernet, redes LAN y otras redes.

- **Capa de paquetes (CORE).** La capa de núcleo es la encargada de desviar el tráfico lo más rápido posible hacia los servicios que han sido solicitados. Es decir, es la capa que conmuta los paquetes, teniendo en cuenta factores como la velocidad, la latencia y los protocolos que permitan converger rápidamente hacia el servicio invocado.

Esta capa tiene conexión con la red de acceso por medio de nodos con switches T160G MPLS.

Dentro de esta capa se encuentran dos componentes importantes como son los gateway troncales y los gateway de señalización.

- Gateway troncales (TG). Son los encargados de convertir la información TDM en información de paquetes y viceversa suministrando las

interfaces E1 y Ethernet para integrar la red PSTN existente con el core de la red multiservicios.

Los gateway utilizados para esta plataforma multiservicios son los ZXSS10 MSG7200 (TG) los cuales están en capacidad de armar y desarmar paquetes de voz para poderlos transmitir a las centrales telefónicas de la red PSTN o a la red de paquetes.

- Gateway de señalización (SG). Tienen la función de traducir la señalización CSN (Circuit Switched Network), conocida como SS7 de la red PSTN existente en señalización de paquetes y viceversa para transmitirlos a través de ambas.

Los gateway de señalización implementados en la red NGN son los ZXSS10 MSG7200 (SG), los cuales funcionan al borde de la red SS7 y la red de paquetes en el modo de circuito conmutado, recibe y envía mensajes SS7 de la red de conmutación de circuitos por medio de E1 y adopta los protocolos de adaptación de capas y transporte como SIP o H.323, realizando la interconexión entre las dos redes internas de señalización. Este gateway también tiene la función de proporcionar rutas de transmisión para la voz.

- **Capa de control.** Como su nombre lo indica en esta capa se realiza el control de todas las operaciones realizadas dentro de la plataforma multiservicios y del control de todos los dispositivos que hacen parte de esta. Asegura el funcionamiento entre las demás capas obteniendo así los mecanismos necesarios para la provisión de los servicios.

Dentro de esta capa reside el softswitch el cual es el elemento central de la red.

El softswitch implementado en la plataforma multiservicios de EMCALI es el ZXSS10 SS1b desempeñando funciones tales como:

- Control de llamadas
- Control y enrutamiento de llamadas y sesiones
- Adaptación de protocolos para interconectar usuarios de diferentes redes
- Provisión de servicios
- Control de los gateway de troncales y de señalización
- Responsable de la tasación del tráfico de voz, de Internet y de todos los usuarios de la red multiservicios

- Administración del ancho de banda y de los recursos (recursos de troncales, servidores de medios, recursos del sistema de procesamiento etc.) del sistema en general, según la demanda de los diferentes servicios
- Establecimiento de canales con diferentes anchos de banda de acuerdo al estado de la red de datos
- Establecimiento de circuitos virtuales requeridos por un patrón de tráfico predeterminado o para el establecimiento de una nueva llamada
- Permitir ofrecer servicios de telefonía clase 5, carrier-class y servicios multimedia
- Control de tráfico IP proveniente de las líneas POTS (a través del modulo NAS del gateway de troncales), de los puertos XDSL de la red NGN o de la red PSTN
- Control de los servicios de Internet
- Control sobre las UAM's, gateway troncales, gateway de señalización y la plataforma de servicios

Este softswitch posee redundancia a nivel de hardware y software. Cuando alguna tarjeta tiene un problema, la tarjeta de backup toma su lugar inmediatamente, guarda un registro de las llamadas y los servicios no son interrumpidos.

A nivel de software el diseño también es redundante, ya que este equipo puede ser configurado en modo de solución redundante, para resolver problemas de seguridad y confiabilidad de red.

En la red multiservicios existen dos softswitches, uno de ellos proporciona el sistema de redundancia como protección contra desastres. Normalmente los dos softswitches controlan los grupos de usuario y los respectivos gateways en modo balanceado de carga. Cuando alguno de los dos softswitch encuentra algún problema, los gateways y unidades de red controlados por este, se vuelven a registrar en el softswitch de backup. De esta forma se pueden seguir prestando los servicios normalmente a los usuarios conservando también, el estado en que se encuentre algún servicio en el momento en que se presente la falla. De esta forma cualquiera de los dos softswitches de la red multiservicios puede trabajar con el 100% de la carga de la red.

• **Capa de Servicios.** Esta última capa es la que contiene cada uno de los servicios que serán utilizados por los usuarios. Estos servicios pueden ser desarrollados por terceras partes en diferentes plataformas sin importar la infraestructura de la red.

La capa de servicios está conformada por la plataforma de servicios ZXUP10, el cual es un servidor de aplicaciones convergente, modular orientado hacia NGN y adopta la arquitectura OSA/Parlay.

El sistema ZXUP10 tiene como principales características:

- Está diseñada para trabajar tanto con la red existente como con la red NGN
- Provee interfaces estándar y abiertas lo cual permite el desarrollo de servicios por terceras partes
- Soporta múltiples protocolos como INAP, CAMEL.WIP, SIP, MGCP y SMPP
- Las aplicaciones pueden ser ejecutadas desde diferentes servidores de aplicaciones reduciendo posibles fallas en el sistema
- Proporciona encapsulamiento de los recursos de la red

• **Componentes de la plataforma ZXUP10.** Dentro de la estructura física de la plataforma ZXUP10 se encuentran elementos como el Parlay Gateway, el servidor de medios, el servidor de aplicaciones, el administrador de operaciones, la consola de mantenimiento (OAM), y el gateway de señalización.

- Parlay Gateway. Es el núcleo del sistema ZXUP10. Encapsula varios niveles de la red de recursos, provee las interfaces API's abiertas de acuerdo al protocolo Parlay y ofrece una plataforma para el desarrollo de servicios por terceras partes. Es el enlace entre las aplicaciones y los elementos de la red. Dentro del Parlay gateway se encuentra el Framework de OSA/Parlay.

Está compuesto por tres módulos funcionales: el módulo de adaptación de protocolos, módulo de funciones básicas y el módulo de control de servicio.

- Módulo de adaptación de protocolos. Selecciona el protocolo adecuado de acuerdo al tipo de red.
- Módulo de funciones básicas. Este módulo es la capa de control y gestión del Parlay Gateway. Aquí están disponibles los SCF's definidos por OSA/Parlay.
- Módulo de control de servicio. Este módulo implementa las API's de Parlay, relacionadas con los servicios de telecomunicaciones y recibe las

invocaciones de aplicaciones de terceras partes permitiendo a estas aplicaciones comunicarse con las otras capas de la red a través de las API's de Parlay.

3. HERRAMIENTAS PARA LA CREACIÓN DE NUEVAS APLICACIONES EN NGN

La arquitectura de las NGN está diseñada para agilizar el desarrollo de nuevos servicios y aplicaciones telemáticos. Como se explicó anteriormente, esta es una de las formas más seguras que tienen las empresas proveedoras de telecomunicaciones de atraer nuevos suscriptores y mantener a los antiguos. Por esa razón se ha favorecido el uso de interfaces abiertas y de software libre para el desarrollo de aplicaciones. A continuación se explican algunas de las herramientas disponibles, estas herramientas incluyen APIS implementados en Java, alternativas que usan XML, scripts entre otras.

3.1 SIP SERVLETS

Las aplicaciones diseñadas utilizando SIP utilizan un conjunto de librerías, implementadas en Java, que brinda herramientas para la creación de servicios sobre redes IP.

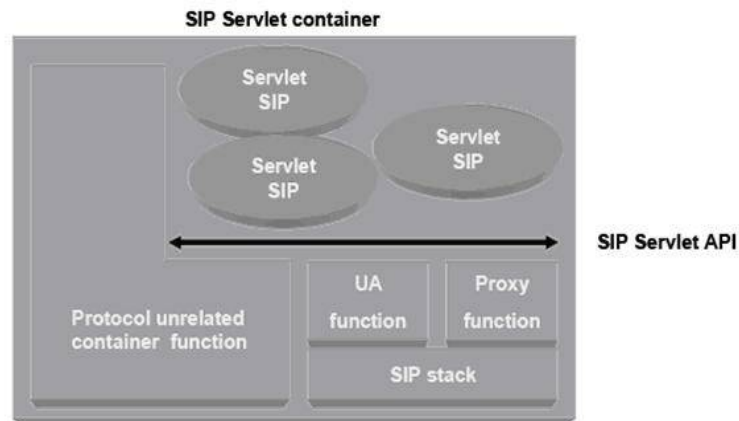
Un SIP servlet es gestionado durante su tiempo de vida por un contenedor de SIP servlet el cual hace parte de un servidor de aplicaciones que proporciona los servicios de red a través de los cuales se reciben y envían las peticiones y las respuestas.²¹

3.2 CCXML (CALL CONTROL EXTENSIBLE MARKUP LANGUAGE)

Basado en XML, adopta una arquitectura basada en red para controlar las llamadas. Este lenguaje realiza el control de las llamadas a través de la respuesta a eventos. Estos eventos generan los cambios de estado en la máquina de estados de las llamadas. CCXML es utilizado para complementar VoiceXML, que no cuenta con la capacidad para controlar algunos tipos de llamadas.

21 Op cit GUEVARA p 47

Figura 5. Contenedor de SIP Servlet



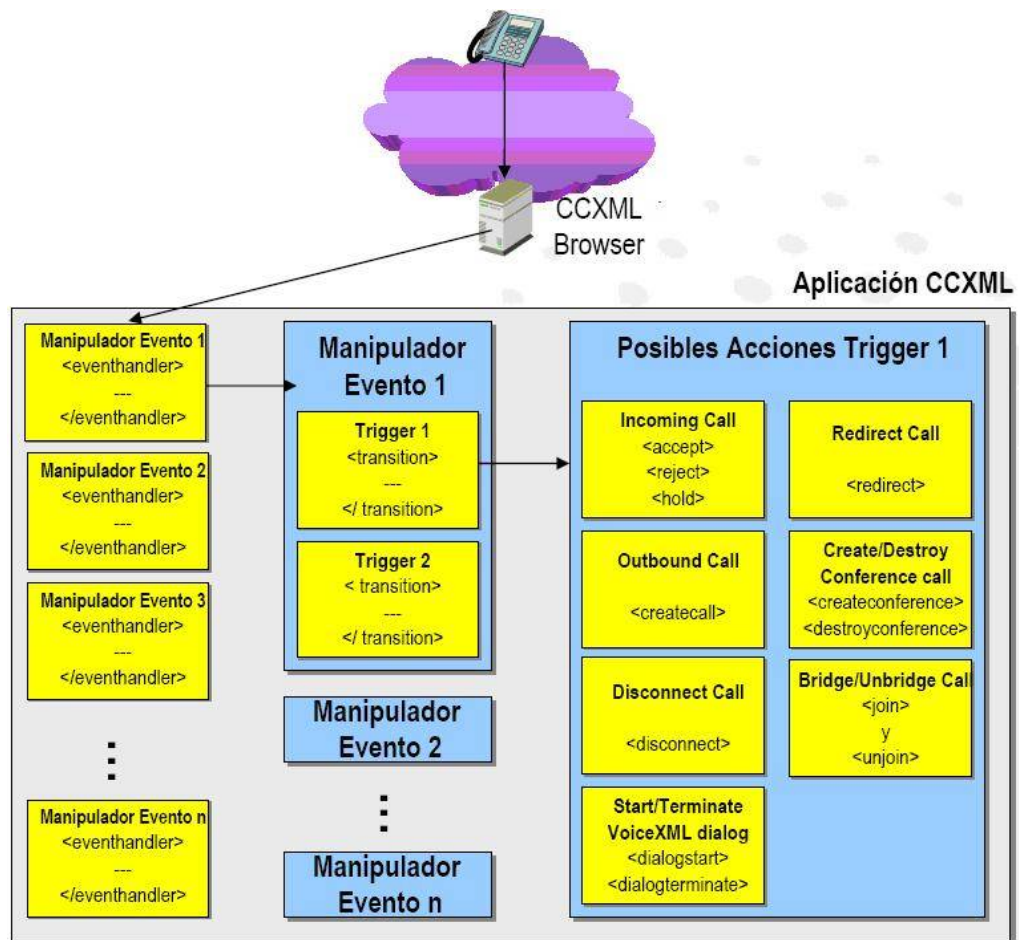
Fuente: GUEVARA, Diana Marcela. Ambientes de desarrollo para la implementación de nuevos servicios telemáticos sobre la red multi-servicios de EMCALI. Pasantía para Ingeniera Electrónica Santiago de Cali: Universidad Autónoma de Occidente. Facultad de Ingeniería. 2008. p 56

Entre las características más importante de CCXML son la facilidad de uso, la flexibilidad como ya se había mencionado anteriormente y la capacidad de procesar aplicaciones complejas. Un ejemplo de una aplicación en CCXML se muestra en la figura 6.

CCXML ofrece las siguientes funciones de control de llamadas:

- Controlar llamadas salientes
- Controlar llamadas múltiples
- “Whisper Transfer”: Envía un mensaje al destinatario antes de establecer una conexión.
- Conferencia
- Gestión de llamadas
- Manipulación de eventos
- Sesión de inicio y finalización del interprete voiceXML.
- Sesión de inicio del intérprete CCXML. Es posible que durante la ejecución de una aplicación se levante otra aplicación.
- Enrutamiento

Figura 6. Modelo de aplicaciones en CCXML



Fuente: GUEVARA, Diana Marcela. Ambientes de desarrollo para la implementación de nuevos servicios telemáticos sobre la red multi-servicios de EMCALI. Pasantía para Ingeniera Electrónica Santiago de Cali: Universidad Autónoma de Occidente. Facultad de Ingeniería. 2008. p 61

3.3 JAVA TELEPHONY API (JTAPI)

Esta es la primera implementación en Java de un API para telefonía, por esta razón es la base de tecnologías más complejas como JAIN y Parlay. Con JTAPI las aplicaciones pueden realizar un control de llamadas, también controlar flujos de audio, también cuenta con herramientas para realizar el cobro por los servicios usados.

JTAPI extendió los esfuerzos realizados por compañías telefónicas que desarrollaron versiones anteriores de APIS para crear servicios en telecomunicaciones (TAPI Y TSAPI). JTAPI es un estándar abierto, y a diferencia de sus predecesores no se encuentra atado a las compañías que la desarrollaron. JTAPI al estar desarrollado en Java cuenta con la ventaja de ser

multiplataforma, porque es el entorno de Java, que se encuentra disponible para varios sistemas operativos, el que se comunica directamente con el sistema operativo.

3.3.1 Descripción de JTAPI. Esta es una interfaz portable, orientada a objetos y desarrollada para Java. Una implementación de JTAPI provee la interfaz entre las aplicaciones de Java y los sistemas de comunicación como los CTI (Integración de computación telefónica).

JTAPI cuenta con un módulo (paquete) core que define un modelo básico de llamada para soportar el establecimiento de llamadas y un número de extensiones principalmente diseñados para modelar funciones de centro de llamadas, llamadas multipartita, enrutamiento de llamadas y demás.²²

La figura 7 muestra el modelo de llamadas de JTAPI. Cada objeto en la figura abstrae una entidad lógica o física de telefonía. Este paquete de núcleo usa los siguientes objetos:

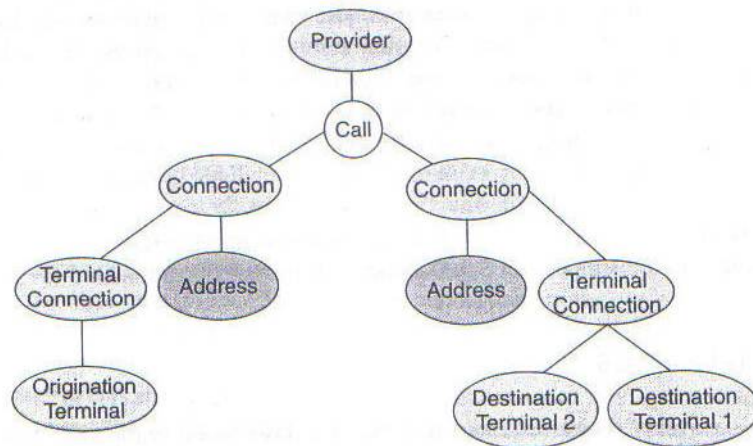
- Proveedor
- Llamada
- Conexión
- Dirección
- Terminal
- Conexión a Terminal

El objeto Proveedor es la abstracción de las funciones ofrecidas por un sistema de telefonía, es decir, es a través de este objeto que JTAPI se relaciona con los servicios relacionados con el subsistema de telefonía.

Un objeto de llamada representa una llamada telefónica entre dos o más terminales. Un objeto de dirección representa el destino lógico, que puede ser una IP o un número telefónico. Los terminales son abstracciones de los dispositivos físicos. Un objeto de dirección puede ser asociado con múltiples terminales.

La Dirección y el objeto de llamada tienen una relación dinámica, que es representada por el objeto de Conexión. Este objeto cuenta con una máquina de estados, el estado de esta máquina depende de la relación entre el objeto de llamada y Dirección.

Figura 7. Modelo de llamada de JTAPI



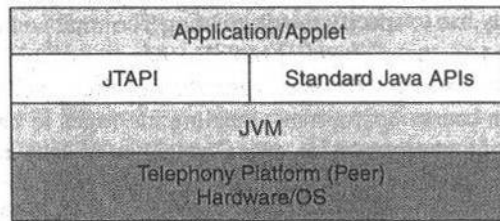
Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 86

JTAPI cuenta con un mecanismo conocido como Capabilities (Capacidades) que le permite a la aplicación obtener las capacidades de un entorno determinado. Estas capacidades pueden ser dinámicas o estáticas. En caso de que una aplicación invoque una capacidad que no se encuentra disponible se arroja una excepción.

También existe la posibilidad de implementar eventos, estas son notificaciones que se envían a la aplicación cuando se producen cambios de estados en los objetos involucrados en la llamada. Para que una aplicación pueda recibir estas notificaciones debe implementar un Listener relacionado con el evento solicitado.

3.3.2 Peers y paquetes de JTAPI. Un peer es la implementación de JTAPI para determinada plataforma de hardware. Es decir, representa el sistema operativo y el hardware.

Figura 8. Implementación de JTAPI y Peers



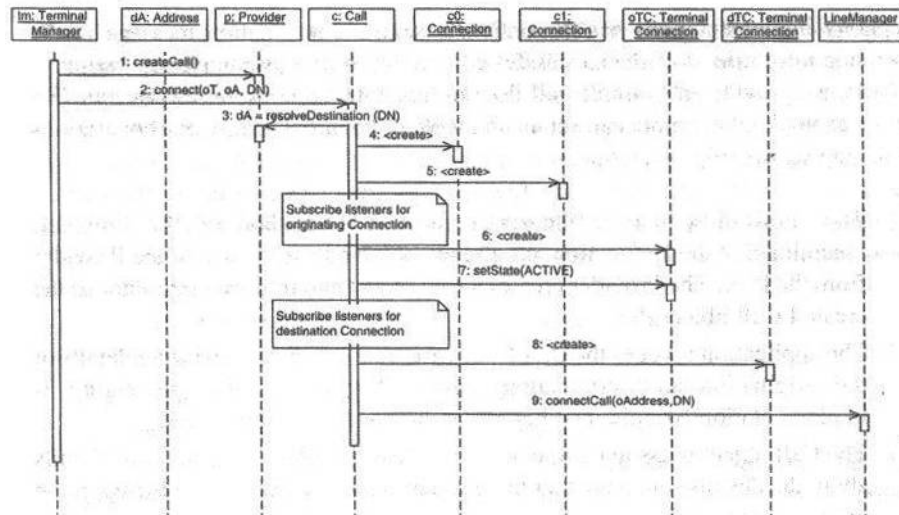
Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 91

JTAPI se encuentra compuesto por los siguientes paquetes:

- Núcleo
- Control de llamadas
- Centro de llamadas
- Móvil
- Teléfono
- Media
- DatosPrivados

• **Núcleo.** Provee las funciones relacionadas con la telefonía básica. Este paquete ofrece métodos esenciales para establecer y terminar llamadas, algunos de estos métodos son: `Call.Connect()`, `TerminalConnection.answer()`, `Connection.disconnect()`, `Provider.createCall()`. La figura 9 muestra un diagrama del establecimiento de una llamada en jtapi

Figura 9. Diagrama de secuencia del establecimiento de una llamada en JTAPI



Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 93

Este proceso sigue los siguientes pasos:

1. La aplicación invoca el método createCall() del proveedor, asumiendo que la aplicación ya ha obtenido una referencia al proveedor. La aplicación recibe una referencia a la llamada.
2. La aplicación invoca Call.connect() especificando los detalles de las partes involucradas.
3. La llamada obtiene, a través del proveedor, detalles acerca del destino de la llamada.
4. El objeto de llamada crea objetos de Conexión, cada uno representando una parte que inicia o termina la llamada.
5. El objeto de llamada crea un objeto llamado TerminalConnection para la parte que inicia la llamada.
6. El objeto de llamada pone el estado del objeto creado en el paso anterior a ACTIVA.
7. El objeto de llamada crea un objeto llamado TerminalConnection para la parte destinataria de la llamada
8. El objeto de llamada contacta un objeto que representa el hardware del teléfono que se encuentra debajo y le solicita mensajes de señalización de llamada.

- **Control de llamadas.** Este paquete hereda del paquete núcleo para proveer funciones avanzadas de control de llamadas. Esto es realizado a través de la implementación de máquinas de estado más compleja, heredando las interfaces y ofreciendo métodos adicionales en cada interfaz involucrada.

Algunas de las funciones ofrecidas por este paquete incluyen Llamada en espera, Transferencia de llamadas, Conferencias.

- **Centro de llamadas.** Este paquete le provee a las aplicaciones la posibilidad de implementar funciones avanzadas que son requeridas en grandes centros de llamadas. Algunas de estas funciones son Distribución automática de llamadas, Enrutamiento de llamadas, Discado predictivo.

- **Móvil.** Este paquete permite realizar aplicaciones para puntos finales inalámbricos. Debido a la antigüedad de este API, está diseñado para redes GSM.

- **Teléfono.** Le ofrece a las aplicaciones control sobre las funciones físicas del teléfono.

- **Media.** Este paquete permite controlar flujos de audio y video.

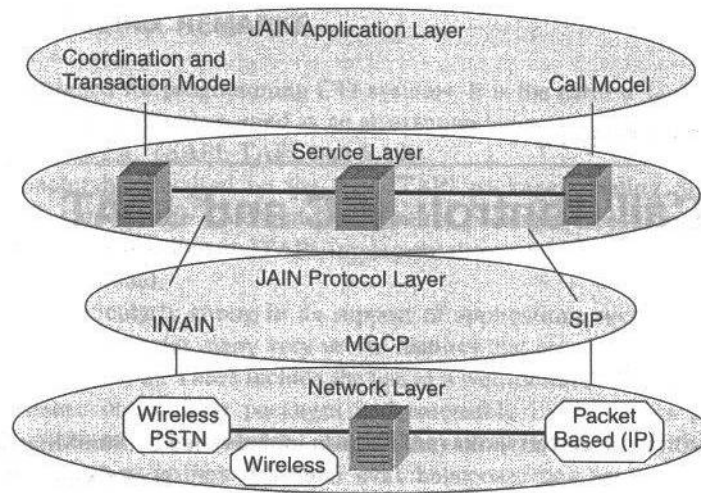
- **Datos privados.** Este paquete le permite a las aplicaciones comunicarse directamente con el hardware debajo. Es una especie de puerta trasera para JTAPI.

3.4 JAIN CALL CONTROL (JCC / JCAT)

JAIN es una iniciativa dentro de la Comunidad de Procesamiento de Java de Sun Microsystems. Esta empezó en 1998 como un foro para el desarrollo de las especificaciones de la tecnología Java.

La arquitectura JAIN, mostrada en la figura 10, posee tres capas especificadas por JAIN: Una capa de red que consiste del hardware físico y las entidades que manejan los protocolos de señalización de bajo nivel como SS7, PSTN, Wireless, IP ó ATM. Una capa de señalización, una capa de servicios (servidores SCP en redes PSTN o Servidores de Aplicación en redes IP). Aplicaciones JAIN, estas son, aplicaciones escritas para usar las funciones del API proveídas por JAIN.

Figura 10. Arquitectura básica JAIN

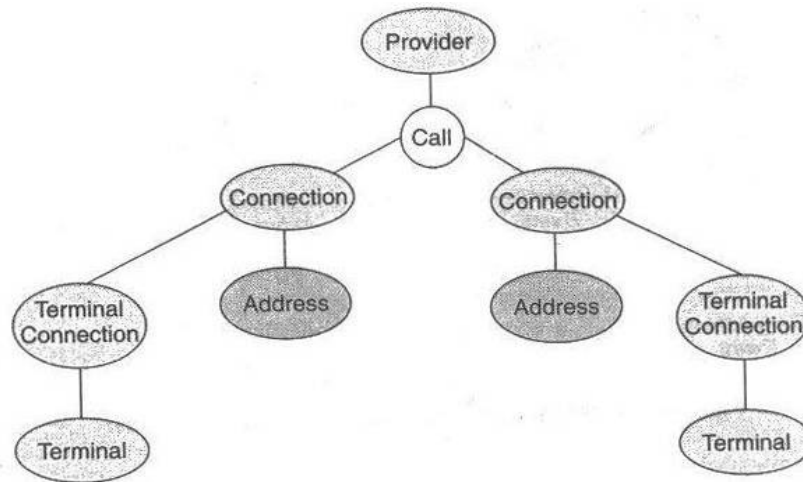


Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 108

3.4.1 Elementos de JCC y capa de aplicación. La arquitectura JAIN permite acceso a la red convergente (NGN) a aplicaciones confiables y no confiables. Las aplicaciones deben pasar antes los puntos de chequeo para el esquema de seguridad implementado. Este esquema de seguridad no sólo autentica y autoriza a las aplicaciones el acceso a la red, sino que también es responsable del proceso inverso.

3.4.2 Control de llamadas en JAIN. Los APIs de JAIN definen una interfaz para las redes de nueva generación, esta interfaz se encuentra especificada orientada a objetos. Esta orientación está diseñada para esconder la arquitectura específica de la red subyacente. Es decir, estos APIs (JCC y JCAT) son independientes de los protocolos de señalización.

Figura 11.Modelo de objetos de JCAT para una llamada entre dos partes



Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 117

JCC y JCAT pueden ser usados para establecer sesiones multimedia a través de las herramientas proveídas por su API. Sin embargo no está diseñado para tener un control completo de estas sesiones, es decir, no se soportan funciones como control de subflujos multimedia o funciones síncronas.

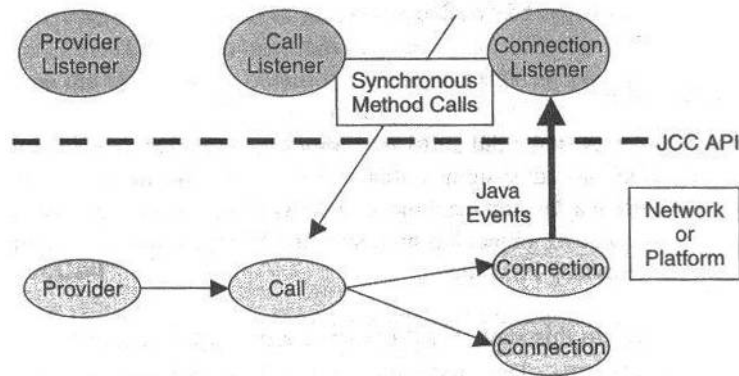
Para el control de llamadas, se asume que la red puede notificarle a la plataforma implementando el API los eventos que ocurran y la plataforma pueda procesar las notificaciones de la forma necesaria para informar a la aplicación. Adicionalmente, la aplicación iniciará acciones que serán traducidas a los protocolos de señalización apropiados.

Las funciones para interacción con el usuario durante una llamada activa son proveídas por paquetes opcionales que están por fuera del alcance de JCC/JCAT.

3.4.3 Escuchadores y fábricas. En JCC/JCAT las aplicaciones se encuentran por encima del API y pueden implementar escuchadores que capturen eventos de interés enviados por la red. La captura de estos eventos le permite a las aplicaciones realizar un control de las llamadas actualmente en pie.

La interacción entre los eventos, los escuchadores y las aplicaciones se muestra en la figura 12.

Figura 12. Patrón de programación usando Java Listeners



Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 117

Este API hace uso de el patrón de diseño comúnmente conocido como fábrica. A través de esa fábrica se obtiene un “peer”. Esto se una implementación específica o propietaria del API.

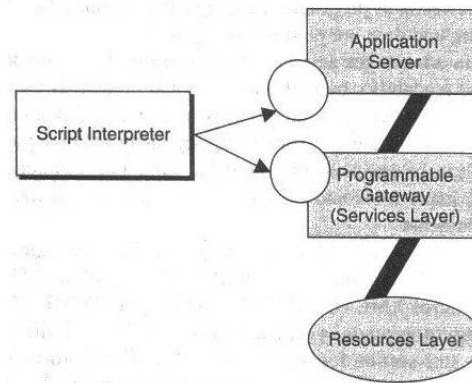
3.5 SERVICE CREATION MARKUP LANGUAGE (SCML)

SCML es un lenguaje de script basado en XML desarrollado dentro del Entorno de Creación de Servicios (SCE) del grupo JAIN. Este lenguaje incluye a otros lenguajes de script que le entregan funciones requeridas para el desarrollo adecuado de aplicaciones para redes de nueva generación, entre estos lenguajes se incluy el CCML.

Los lenguajes de script tienen diversas ventajas que los hacen una alternativa atractiva entre ellas se incluye: son livianos, altamente personalizables, típicamente interpretados y apropiados en el área de desarrollo rápido de aplicaciones. A pesar de esto, no son tan flexibles y robustos como los lenguajes de programación.

3.5.1 Desarrollo e implementación de un script SCML. SCML es utilizado para crear aplicaciones que hacen uso de una capa de servicios. Como estos lenguajes son interpretados, requieren de un agente que se encargue de realizar la traducción Los scripts pueden ser almacenados en bases de datos generales o en bases de datos de suscriptores como HLRS. La figura 13 muestra la relación del interprete, la capa de aplicación y el Gateway.

Figura 13. Papel del intérprete en SCML



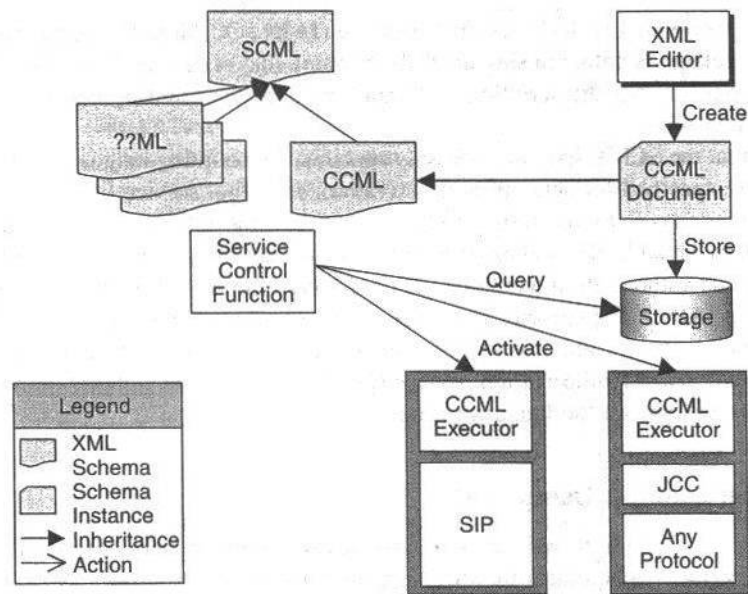
Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 228

Los scripts pueden ser utilizados para realizar control de llamadas de primeras o terceras personas, pueden ser invocados por un evento de la red o durante el procesamiento básico de llamada. También pueden controlar la llamada basados en información de otras interfaces o configuraciones externas. La figura 14 muestra la creación, implementación y ejecución de un script.

Los scripts pueden ser levantados a través de determinado criterio, este criterio es verificado por un validador. Este criterio puede ser un disparador de AIN, ó un evento Java en la plataforma de procesamiento de llamadas.

La especificación SCML sólo especifica el lenguaje de script, no se especifican mecanismos para la activación, desactivación y otros eventos basados en el ciclo de vida del servicio. Es recomendado que antes de que se ejecute un script determinado una entidad se encargue del proceso de autentificación y autorización.

Figura 14. Creación, implementación y ejecución de un script



Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 229

3.6 OSA/PARLAY

El grupo Parlay es un consorcio formado para desarrollar interfaces de programación de aplicaciones de tecnología independiente que permiten el desarrollo de aplicaciones para redes convergentes. Parlay integra redes multimedia de internet y redes inteligentes con aplicaciones IT a través de una interfaz segura y confiable que ha sido ampliamente utilizada en redes de telecomunicaciones globalmente.²³

El acceso abierto a servicios (OSA) es parte de los trabajos de 3GPP y describe como los servicios son diseñados en una red UMTS y redes convergentes.

OSA/Parlay define una arquitectura que habilita el trabajo entre las aplicaciones de telecomunicaciones y las características de las redes actuales a través de una interfaz abierta y estandarizada. Las funciones de la red son abstraídas en forma de Funciones de Capacidad de Servicio (SCF's). Adicionalmente, gracias a su esquema de seguridad, se facilita el desarrollo de aplicaciones por parte de terceras personas.

²³ MUÑOZ O Mario. SCMM Metamodelo para la creación de aplicaciones en Redes de Siguiete Generación. Tesis doctoral para Ingeniero en Comunicación. Leganés: Universidad de Carlos III de Madrid. 2003. p 15

Este API se encuentra instalado en la capa de aplicaciones de la arquitectura NGN explicada anteriormente. EMCALI cuenta con un servidor de aplicaciones que tiene implementado OSA/Parlay.

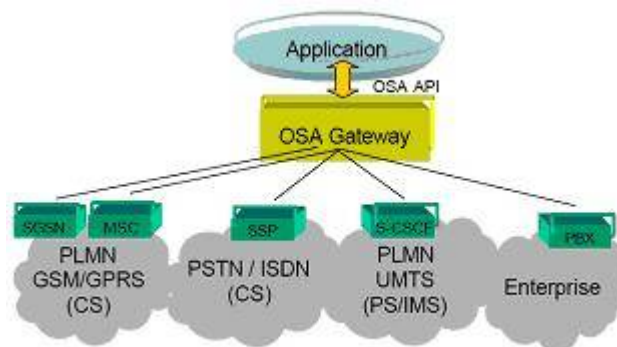
Las ventajas del uso de APIs de OSA/Parlay son:

- Esconde la complejidad de la red, los protocolos y su implementación específica.
- Es apropiado para el desarrollo de aplicaciones por terceras personas. A pesar de requerir poco conocimiento en telecomunicaciones, usa herramientas de programación populares lo que facilita el desarrollo de aplicaciones.
- Provee un acceso seguro y controlado a las capacidades de la red. Esto a través de un marco que asegura que sólo las aplicaciones autorizadas pueden acceder a la red.
- Expone casi todas las capacidades de la red provistas por el protocolo de comunicaciones correspondiente y facilita el desarrollo de servicios combinando varias capacidades de servicio e integrando aplicaciones IT.
- La madurez de OSA/Parlay se ha vuelto significativa después de la creación de un grupo de trabajo conjunto que incluye 3GPP, ETSI, Parlay y JAIN.²⁴

Las API's de Parlay cumplen una función fundamental en las redes NGN ya que cumplen funciones de puertas de enlace para distintas redes hacia la red NGN. Es decir, estas funciones comunican la capa de aplicación con las capas de transporte y acceso en una plataforma NGN. La figura 15 muestra esta topología

24 Op cit FALCARIN

Figura 15. OSA/Parlay como Gateway en una red NGN



Fuente: CHNEPS-SCHNEPPE Manfred, UMITRY AbavaNet. "Parlay Architecture and NGN" [en línea] Disponible en internet http://www.abavanet.ru/download/1-4-3_schneps.pdf

3.6.1 Componentes de OSA/Parlay. Los componentes de una arquitectura OSA/Parlay son:

- **Servidores de aplicaciones.** Son los encargados de ejecutar las aplicaciones que darán servicio a los usuarios mediante el control de la funcionalidad subyacente en la red de telecomunicaciones usando las capacidades de servicio.
- **Servidores de capacidades de servicio.** Implementan las capacidades de servicio o puertas de enlace, éstos mediante interfaces abiertas permiten a las aplicaciones ejecutar la funcionalidad de los elementos de la red.
- **Plataforma.** También es conocida como "framework" permite que las aplicaciones descubran las capacidades de servicio de forma segura.
- **Elementos de la red de telecomunicaciones.** Agrupan los nodos internos de los distintos tipos de redes.

3.6.2 Capacidades de servicio en OSA/Parlay. Estas capacidades de servicio, son las funciones principales de la API de OSA/Parlay. La versión actual de esta API es la 6. La tabla 1 muestra las capacidades de servicio disponibles en la última especificación de esta API. La parte uno es una descripción general de las posibilidades de la API y no es una función como tal.

A continuación se describen de forma general las capacidades de servicio de OSA/Parlay se describirán en detalle en el siguiente capítulo.

La tabla 1 muestra las capacidades de servicio disponibles en la versión actual del API de Parlay.

Tabla 1. Capacidades de Servicio en OSA/Parlay

Capacidad del servicio
Descripción general
Definiciones de datos comunes
Framework (Plataforma)
Control de llamadas
Interacción del usuario
Movilidad
Capacidades del terminal
Control de sesiones de datos
Mensajería general
Administrador de conectividad
Administración de cuentas
Tarifación
Administración de políticas
Administración de presencia y disponibilidad
Mensajes multimedia
Escuchar un servicio

• **Framework (Plataforma).** Es encargada de autorizar a las aplicaciones el acceso a los servicios. Para este proceso se tienen los siguientes pasos:

- Autenticación.
- Registro
- Descubrimiento de servicios
- Firma del acuerdo de servicio

• **Control de Llamadas.** Permite el control de las llamadas que se produzcan en la red, permite desde establecer llamadas básicas entre dos usuarios hasta manipular llamadas multipartita de multimedia. Se definen cuatro tipos de control: Control de llamadas genérico, control de llamadas multipartitas, control de llamadas multimedia, control de llamadas de conferencia.

• **Interacción de Usuario.** Es usado por una aplicación para interactuar con los usuarios finales. Provee funciones para enviar información o, reunir información de ellos. Puede también iniciar una sesión de interacción con un usuario que no está participando en la llamada.

• **Movilidad.** Tiene dos APIs separadas: Estatus de terminal móvil y ubicación

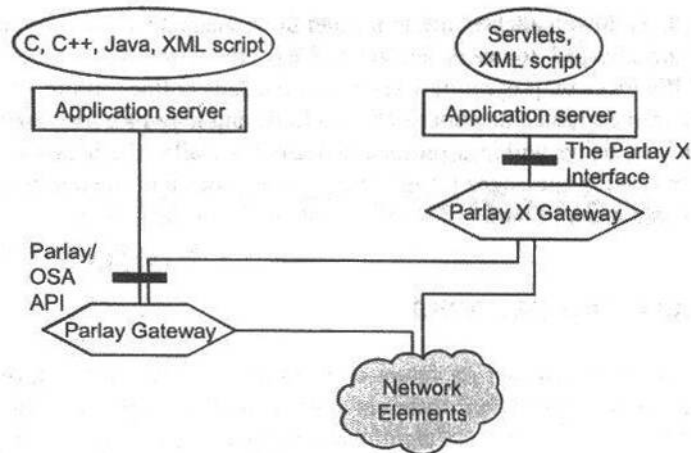
de terminal móvil. El API de estado permite a una aplicación obtener el estatus de usuarios de telefonía IP fijos, móviles

- **Capacidad de Terminal.** Le permite a una aplicación obtener las capacidades conocidas de un terminal de usuario final.
- **Control de sesiones de datos.** Controla los servicios de datos de valor agregado que le permiten a la información ser enviada y recibirla a través de la red telefónica.[16]
- **Mensajería General.** Soporta la administración de buzones y le permite a las aplicaciones enviar, almacenar, recuperar y recibir mensajes.
- **Administración de Conectividad.** Le permite a las aplicaciones especificar los parámetros de QoS para los paquetes viajando a través de la red de los proveedores de red. Puede ser usado para establecer y quitar el tubo virtual de QoS.
- **Administración de cuentas.** Le permite a las aplicaciones solicitar balances de cuentas, recoger historias de transacción y habilitar o deshabilitar notificaciones de eventos relacionadas con cobros.
- **Cobro.** Permite a las aplicaciones cobrar a las partes por los servicios ofrecidos.
- **Administración de Políticas.** Es utilizado para crear, actualizar y ver políticas. Le permite a las aplicaciones subscribir eventos relacionados con políticas de servicio y obtener estadísticas de eventos.
- **Administración de Presencia y Disponibilidad.** Le permiten a las aplicaciones obtener y poner información acerca de la presencia y disponibilidad de un usuario.

3.6.3 Parlay x. Este es un producto desarrollado en XML, permite a los programadores desarrollar aplicaciones que funcionen con la filosofía de OSA/Parlay, aunque con algunas restricciones.

La figura 16 muestra la relación entre los Servicios Web de Parlay X y los API's de Parlay.

Figura 16. Relación entre Parlay X y los API's de Parlay



Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 233

• **Funciones de Parlay X** . Las funciones disponibles para Parlay X son las siguientes:

- Establecimiento de llamadas por parte de las aplicaciones.
- Tratamiento por parte de las aplicaciones de llamadas originadas en la red de telecomunicaciones subyacente.
- Envío de mensajes SMS.
- Envío de mensajes multimedia.
- Tarifación
- Gestión de cuentas de usuario.
- Consulta del estado de los usuarios.
- Localización de los terminales

3.6.4 Parlay client hub (pch)

El grupo OSA/Parlay diseñó un API general para el desarrollo de aplicaciones sobre redes NGN que utiliza Java fundamentalmente como lenguaje de programación. En este API se establece una estructura segura y abierta para el acceso de aplicaciones autorizadas a los recursos y al control de llamadas, de una o varias partes. Llamadas de datos, voz o multimedia. Sin embargo, a partir de estas implementaciones se han creado niveles de abstracción mayor que van orientados a facilitar el proceso de desarrollo de aplicaciones o ser compatibles con determinada tecnología.

Este caso es el de dos niveles de abstracción de OSA/Parlay, el primero es Parlay X. Este estándar está soportado en XML, y son soportados por la infraestructura Web, por ejemplo HTTP.

El segundo caso es del Parlay Client Hub, desarrollado por la empresa estatal china ZTE e implementado en los softswitch y plataformas de aplicaciones propietarias. Esta implementación será explicada porque forma parte de los alcances de este proyecto y la aplicación desarrollada usa esta plataforma, sin embargo se omitirá información confidencial propia de ZTE.

Parlay Client Hub se encuentra instalado en la plataforma de aplicación de la red multiservicios de EMCALI. Esta herramienta se encuentra integrada con el entorno de Java funcionando en ese equipo (JRE 1.3) y el sistema operativo AIX.

Esta herramienta se encuentra por encima del nivel de los servicios y del API de OSA/Parlay y por debajo de las aplicaciones. Es decir, todas las interfaces y herramientas del API Parlay, se encuentran implementadas de determinada manera e integradas con las bases de datos instaladas en el servidor de aplicaciones. Esta abstracción de PCH es empaquetada en forma de paquetes jar, instalados en la plataforma de aplicación.

Esta abstracción facilita el desarrollo de aplicaciones porque se ocupa de tareas administrativas y del proceso de autenticación, descubrimiento y obtención de servicios. Igualmente, los métodos que se encuentran en el API de Parlay se encuentran implementados para ser compatibles con la tecnología de la plataforma de servicios.

De esta forma, el desarrollo de aplicaciones se realiza interactuando con las interfaces y clases disponibles en PCH y no directamente con las ofrecidas por el API de OSA/Parlay. Esto libera a los desarrolladores de realizar implementaciones de CORBA.

A nivel operacional, PCH y las aplicaciones se encuentran en el mismo proceso, lo que incrementa la eficiencia de ejecución.

- **Funciones de PCH.**

- Manejo de sesiones. PCH maneja las sesiones de todas las aplicaciones ante el Framework. A través de sesiones PCH recibe llamados de capacidades de servicios y los distribuye a las diferentes aplicaciones de cliente. Adicionalmente, las aplicaciones cliente usan sesiones para invocar APIs proveídas por las capacidades de servicio.
- Administración de aplicaciones. PCH provee servicios comunes para aplicaciones cliente.
- Manejo de hilos. Para incrementar la eficiencia de operaciones, PCH tiene un nuevo hilo para procesar los llamados a capacidades de servicios. El número de hilos que se pueden manejar es configurado

basado en las condiciones reales.

- Conectividad con la base de datos. Las aplicaciones cliente hacen consultas a las bases de datos durante la lógica de sus operaciones. Para incrementar la velocidad de conexión y mejorar el funcionamiento del sistema, PCH provee una conexión configurable con la base de datos. La cantidad de conexiones máximas y mínimas son configuradas según las condiciones reales.

En el caso de la red multiservicios de EMCALI, el sistema cuenta con una base de datos Oracle, que es configurada usando sus respectivas herramientas.

- Sesiones en PCH. PCH definió cuatro interfaces de sesión relacionadas con el control de llamadas. Estas interfaces están basadas en el control de llamadas del API de Parlay. Estas sesiones son: Session, MPSession, MMSession y CCSession.

Session provee métodos relacionados con el control de llamadas ordinarias. MPSession hereda de Session y provee métodos relacionados con el control de llamadas multipartita. MMSession hereda de MPSession y provee métodos relacionados con el control de llamadas multimedia. CCSession hereda de MMSession y provee métodos relacionados con el control de llamadas de conferencia.

PCH asocia los ID de sesión de todas las capacidades de servicio relacionadas para llamar a una sesión. PCH también administra las sesiones de todas las aplicaciones, incluyendo crear sesiones, actualizar el estado de la sesión y borrar sesiones.

- Archivos de configuración y logs. PCH se encuentra implementado para tener mucha flexibilidad en el desarrollo de servicios. Una forma de implementar esto es a través de aplicaciones altamente parametrizables. Esto se hace a través de archivos de configuración con extensión .properties. De esta forma se le puede añadir la funcionalidad a un servicio o adaptarlo a otras condiciones sin necesidad de reescribir parcialmente el código, solamente modificando el archivo de configuración.

PCH implementa una forma de informar sobre los eventos y alarmas que se producen durante la ejecución de una aplicación, esto se hace a través de archivos de log, que se encuentran formateados según un archivo llamado log4j.properties.

- PCH-OAM. PCH y la plataforma de aplicaciones de la red multiservicios de EMCALI cuentan con una consola de administración, que permite configurar de forma offline los servicios para que sean reconocidos por el Framework.

Como se mencionó anteriormente, el Framework debe conocer de alguna forma offline, que aplicaciones se encuentran autorizadas para acceder a las capacidades de servicio de la red. Esta consola permite este proceso y también configurar la ubicación de los archivos de properties, logs y la clase lanzadora del proceso.

OAM se comunica directamente con la base de datos y almacena allí los datos configurados por los desarrolladores al momento de cargar la aplicación. El Framework y las aplicaciones leen los datos escritos previamente en la base de datos.

4. OSA/PARLAY

El grupo Parlay es un consorcio formado para desarrollar interfaces de aplicación, estándares y abiertas, de tecnología independiente y que permitan el desarrollo de aplicaciones para redes convergentes. Integra redes multimedia y redes inteligentes con aplicaciones telemáticas que utilizan una interfaz segura y confiable para acceder a los recursos de la red.

El acceso abierto a servicios (OSA) es parte de los trabajos de 3GPP y describe como los servicios son diseñados en una red UMTS.

OSA/Parlay es un API que ha ganado popularidad gracias a las ventajas que ofrece, entre ellas, la posibilidad de desarrollar servicios que combinen varias capacidades de las redes subyacentes y su marco de seguridad, conocido como Framework. La red multiservicios de EMCALI, cuenta con una plataforma de aplicaciones que implementa OSA/Parlay, el servicio diseñado e implementado en este trabajo fue desarrollado utilizando OSA/Parlay.

Utilizar el API de OSA/Parlay presenta múltiples ventajas, las ventajas mostradas a continuación se verán fundamentadas por el análisis al modelo propuestop or OSA/Parlay:

- Esconde la complejidad de la red, los protocolos y su implementación específica.
- Es apropiado para el desarrollo de aplicaciones por terceras personas. A pesar de requerir poco conocimiento en telecomunicaciones, usa herramientas de programación populares lo que facilita el desarrollo de aplicaciones.
- Provee un acceso seguro y controlado a las capacidades de la red. Esto a través de un marco que asegura que sólo las aplicaciones autorizadas pueden acceder a la red.
- Expone casi todas las capacidades de la red provistas por el protocolo de comunicaciones correspondiente y facilita el desarrollo de servicios combinando varias capacidades de servicio e integrando aplicaciones IT.
- La madurez de OSA/Parlay se ha vuelto significativa después de la creación de un grupo de trabajo conjunto que incluye 3GPP, ETSI, Parlay y JAIN.²⁵

²⁵ Op cit, FALCARIN

4.1 HISTORIA DE OSA/PARLAY

Los orígenes de el API desarrollado por OSA/Parlay pueden rastrearse hasta el modelo que lo influenció altamente, este modelo es conocido como TINA-C (Telecommunications Information Architecture Consortium). Este consorcio fue una de las primeras organizaciones que diseñó una arquitectura formal de servicio.

TINA definió dos principios fundamentales para separación. El primero fue que la lógica de procesamiento de la llamada estuviera separada de la lógica de los servicios, este concepto fue introducido por primera vez con la introducción de AIN. El segundo concepto realiza una diferenciación entre el acceso al servicio y el uso del servicio. Este segundo principio es una de las características fundamentales de OSA/Parlay y evolucionaría en el concepto de Framework.

El acceso y uso de los servicios con conceptos realizados en términos de sesiones. Las aplicaciones encuentran servicios y solicitan el uso de ellos. El uso de servicio incluye el control del comportamiento del servicio e intercambio del contenido del servicio.

El beneficio fundamental de este concepto de sesión es que los procesos AAA (Autenticación, autorización y accounting) junto con funciones adicionales como balanceo de carga son centralizados y pueden ser reutilizados por otros servicios.

Otra ventaja con el concepto de sesiones es que los servicios pueden ser desarrollados independientemente de los aspectos de acceso a los servicios. De esta forma, es posible que el desarrollo de nuevos servicios sea realizado por terceras personas.

El resultado entregado por TINA-C consiste en una serie de especificaciones arquitecturales y modelos de información de negocios. El Object Management Group (OMG) tomó el trabajo realizado por TINA-C y produjo una especificación de interfaz, esta interfaz es conocida como TSAS (Telecommunications Service Access and Subscription).

El grupo de Parlay fue fundado en 1998, después de la finalización de la fase 1 de TINA, que tuvo lugar a finales de 1997. Estaba constituido inicialmente por cinco compañías: British Telecom, Microsoft, Nortel Networks, Siemens y Ulticom. La primera especificación liberada fue la versión 1.2 esta versión sólo incluía las funciones de servicio de Control Genérico de Llamadas, Interacción General de Usuario y Administración de Conectividad.

Hasta ese momento, Parlay era un grupo cerrado y no compartía documentos con compañías que no fueran miembros. Al mismo tiempo de la publicación de la versión 1.2 el grupo 3GPPP (3rd Generation Partnership Project) creó un grupo Ad-hoc para Acceso Abierto a Servicios (OSA).

En 1999 varias compañías se unieron al grupo Parlay y este cambió a ser una organización abierta, entre las compañías figuran: AT&T, Cisco, Cegetel, Ericsson, IBM y Lucent Technologies. Adicionalmente el 3GPP desarmó el grupo Adhoc de OSA y creó un grupo dedicado de OSA que trabajaba dentro del Grupo de Especificaciones Técnicas para Redes Núcleo (TSG CN por sus siglas en inglés).

En el 2001, se llevó a cabo un acuerdo que permitió un desarrollo significativo de Parlay, el grupo Parlay firmó un convenio con la ETSI y juntos colaboraron en el desarrollo y publicación de las especificaciones para las Interfaces para Programar Aplicaciones para Acceso Abierto a Servicios (OSA). Con este convenio se conformó el grupo de Trabajo Conjunto (Joint Working Group JGW) que es el responsable del desarrollo de especificaciones para OSA y Parlay. Posteriormente, la ITU-T reconoció los esfuerzos realizados por el JGW.

La versión actual de OSA/Parlay es la versión 5, aunque actualmente está en desarrollo la versión 6. Con la publicación de estas versiones el estándar OSA/Parlay ha adquirido cierta madurez, que es confirmada por las pruebas de interoperabilidad y las implementaciones comerciales que se han realizado hasta el momento, entre ellas la implementación realizada por ZTE para las redes NGN.

4.2 MODELO CONCEPTUAL DE OSA/PARLAY

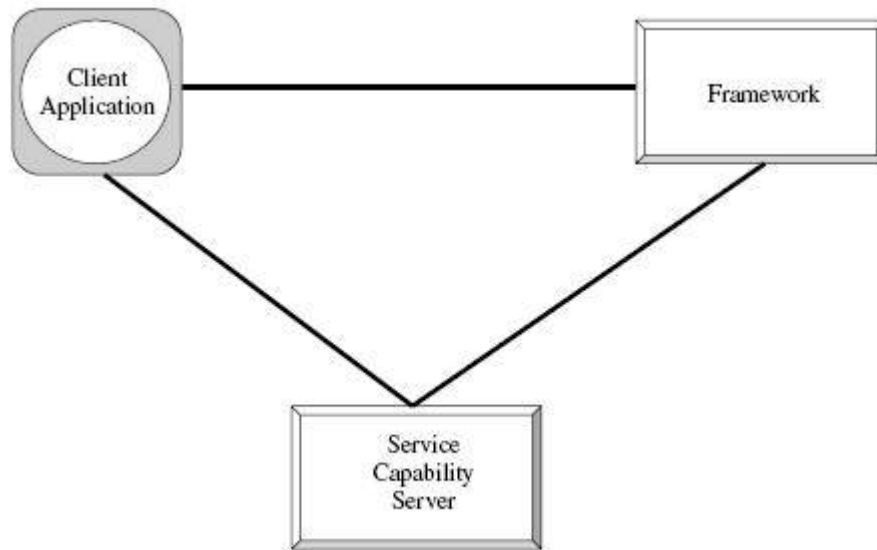
El objetivo de OSA/Parlay es proveer un mecanismo para que las aplicaciones cliente puedan tener acceso a las funcionalidades (Capacidades de Servicio) disponibles en las redes de telecomunicaciones.²⁶ Para lograr este objetivo OSA/Parlay tiene un modelo conceptual que recoge las experiencias positivas de modelos anteriores como el propuesto por AIN y TINA-C y las complementa con un esquema de seguridad conocido como Framework.

OSA/Parlay define una arquitectura compuesta de una interfaz abierta y estandarizada, como hemos visto anteriormente, a pesar de ser un grupo abierto el JWG es un grupo que cuenta con el aval de la ITU-T en los procesos de desarrollo de OSA/Parlay.

4.2.1 Componentes de OSA/Parlay. Fundamentalmente son tres las entidades que interactúan en OSA/Parlay. La figura 17, muestra un resumen de ellos

26 Op cit, UNMEHOPA p 61

Figura 17. Triángulo de OSA/Parlay

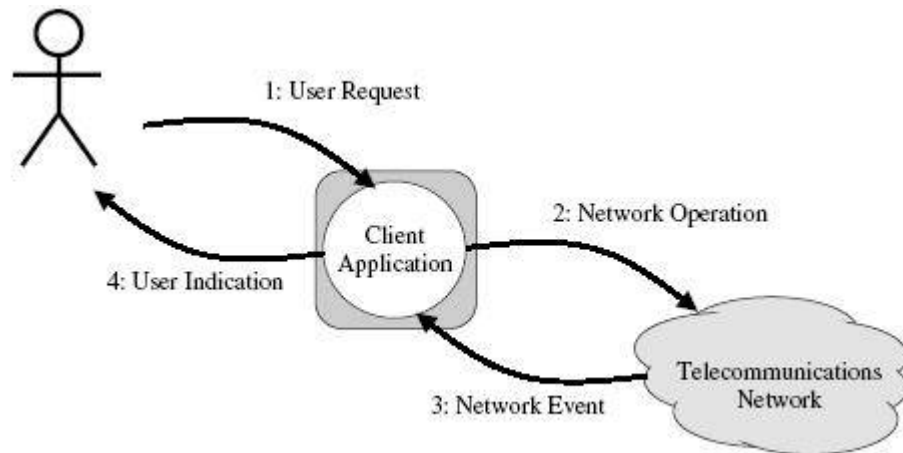


Fuente: UNMEHOPA Musa, VEMURI Kumar, BENNET Andy. Parlay/OSA: From Standards to Reality. Willey 2006. p 62

- **Las aplicaciones Cliente.** Estas aplicaciones residen en la plataforma de aplicaciones de la red NGN y hacen uso de las funcionalidades de la red. A estas aplicaciones los clientes realizan solicitudes. Las aplicaciones toman las solicitudes de los usuarios y las convierten en operaciones de red, igualmente toman los eventos de la red y los convierten en notificaciones a los usuarios. La figura 18 muestra esta interacción:

- **Capacidades de Servicio (SCS).** Estas capacidades son abstracciones de las funciones que puede brindar determinada red a una aplicación. De esta forma, las capacidades de servicio realizan la traducción de las órdenes de la aplicación a operaciones en la red. De esta forma, la aplicación sólo necesita comprender como funciona la capacidad de servicio y no la red. Incluso, es posible que una sola SCS implemente, según el tipo de red, de forma distinta la misma operación. De esta forma una aplicación utilizando la función de control de llamadas, puede ser capaz de controlar datos de redes móviles, PSTN, o VoIP.

Figura 18. Interacción entre aplicación y Red.



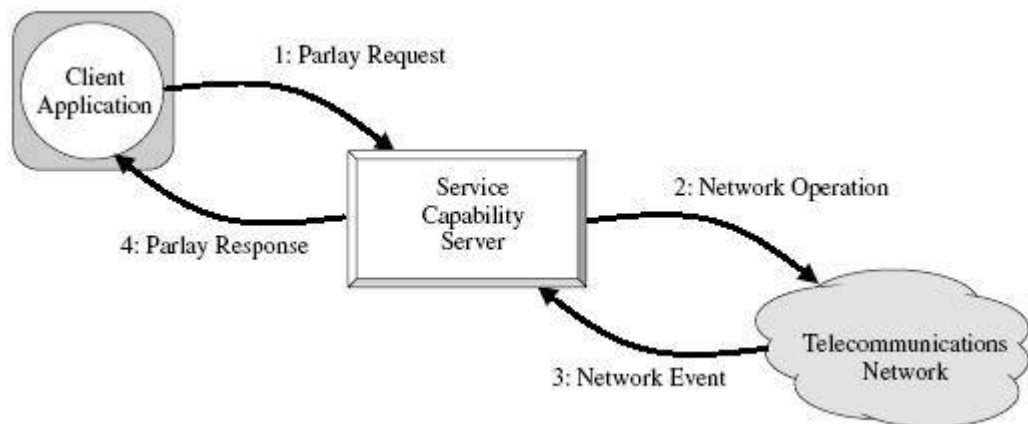
Fuente: UNMEHOPA Musa, VEMURI Kumar, BENNET Andy. Parlay/OSA: From Standards to Reality. Willey 2006. p 62

Como es posible que una función invocada en una aplicación consista en varias operaciones dentro de determinada red, y que estas operaciones pueden tomar tiempo en realizarse y alguna de ellas puede fallar, OSA/Parlay implementa un tipo de métodos que pueden ser asíncronos, es decir, métodos que le retornan inmediatamente el control a la aplicación, pero que posteriormente invocan un método que indica si se ha realizado exitosamente la operación o si ha fallado. Este tipo de métodos se explica más adelante en este capítulo.

Los SCS también cuentan con mecanismos para implementar políticas de aplicación de los servicios que brindan, de esta forma es posible autorizar a las aplicaciones para que utilicen un servicio de forma parcial. La interacción entre los SCS y la red es mostrada en la figura 19.

- **Framework.** Es la entidad encargada de autorizar inicialmente a los servicios el uso de las funcionalidades de la red. También se encarga de autorizar que las aplicaciones tengan acceso a los servicios. Normalmente el Framework reside en la empresa proveedora de telecomunicaciones. Esta entidad es una de las características fundamentales de OSA/Parlay. El interés por implementar un esquema de seguridad efectivo reside en que las aplicaciones pueden tener un control directo sobre las llamadas que se realicen en la red, de esta forma, aplicaciones maliciosas pueden realizar múltiples ataques como DoS. Este Framework autoriza a las aplicaciones y los servicios basándose en acuerdos de servicio previamente firmados entre las empresas proveedoras de telecomunicaciones y los proveedores de servicios o desarrolladores de aplicaciones. Igualmente, las aplicaciones cuentan con mecanismos para autenticar el Framework.

Figura 19. Interacción entre el SCS y la red



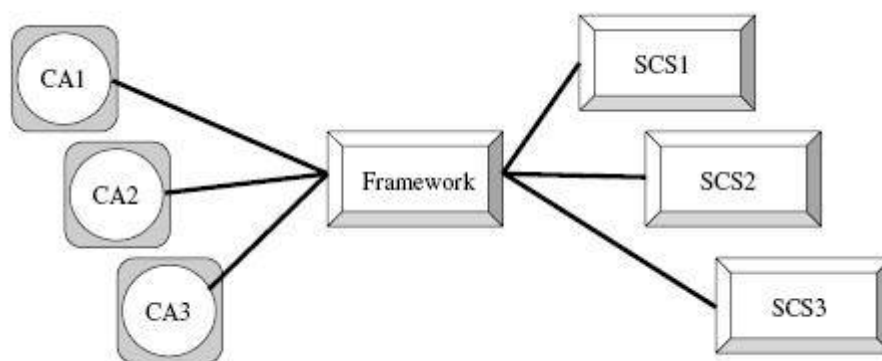
Fuente: UNMEHOPA Musa, VEMURI Kumar, BENNET Andy. Parlay/OSA: From Standards to Reality. Willey 2006. p 64

La figura 20, muestra al Framework como un punto de comunicación inicial entre los servicios y las aplicaciones:

4.3 PATRONES DE DISEÑO Y CONVENCIONES DE NOMBRES

Al ser un estándar abierto OSA/Parlay define algunos patrones de diseño y convenciones de nombres. Estos patrones son característicos de las interfaces ofrecidas y se encuentran justificados por la arquitectura de OSA/Parlay.

Figura 20. Framework como punto de contacto entre SCS y aplicaciones cliente



Fuente: UNMEHOPA Musa, VEMURI Kumar, BENNET Andy. Parlay/OSA: From Standards to Reality. Willey 2006. p 70

4.3.1 Tipos de Interfaces. Es a través de las interfaces en OSA/Parlay que se abstrae la funcionalidad de la red. Los tipos de interfaz son los siguientes:

- **Interfaces del framework hacia las aplicaciones.** Son utilizadas para iniciar sesiones de acceso a los servicios, por medio de mecanismos básicos como autenticación, autorización, descubrimiento, selección del servicio y acceso al servicio permiten hacer uso de los SCF's. Estas interfaces se encuentran en el framework y son usadas por la aplicación. Pueden ser identificadas por medio del prefijo "Ip<name>".

- **Interfaces de callback que corren del lado de la aplicación hacia el framework.** Estas interfaces necesitan ser implementadas por una aplicación (cliente), son utilizadas para responder a eventos. Los nombres de estas interfaces tienen el prefijo "IpApp<name>"

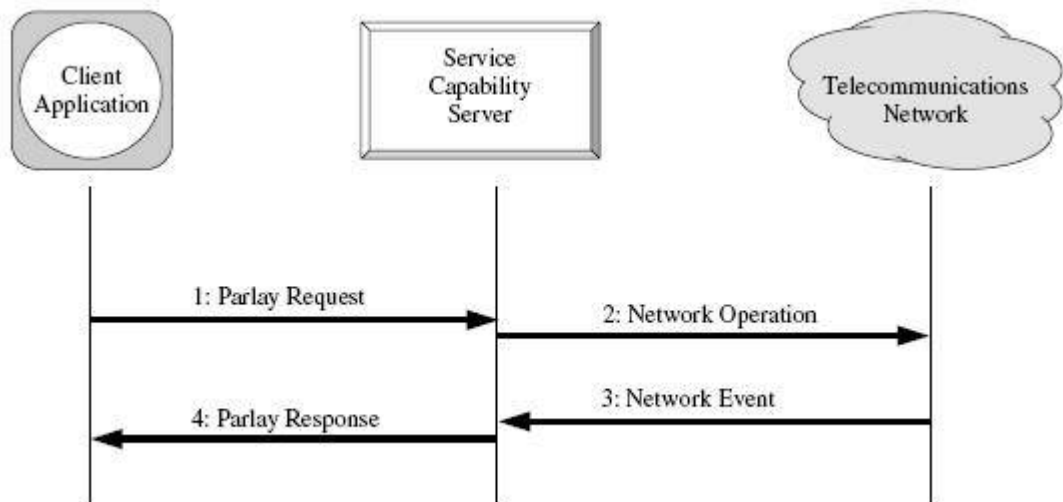
- **Interfaces del framework que son utilizadas por los servicios.** Por medio de estas interfaces los servicios pueden ser registrados en el framework. El prefijo utilizado para identificar estas interfaces es "IpFw<name>"

- **Interfaces de un servicio que son utilizados por el framework.** El nombre de este tipo de interface de servicio tiene el prefijo "IPSvc<name>"

- **Interfaces del lado de las aplicaciones y de los SCF's que son compartidas.** Estas últimas pueden ser servicios individuales los cuales son solicitados por el cliente permitiendo la ejecución de aplicaciones desarrolladas por terceros a través de la interface. El nombre de este tipo de interface tiene el prefijo "IpClient"

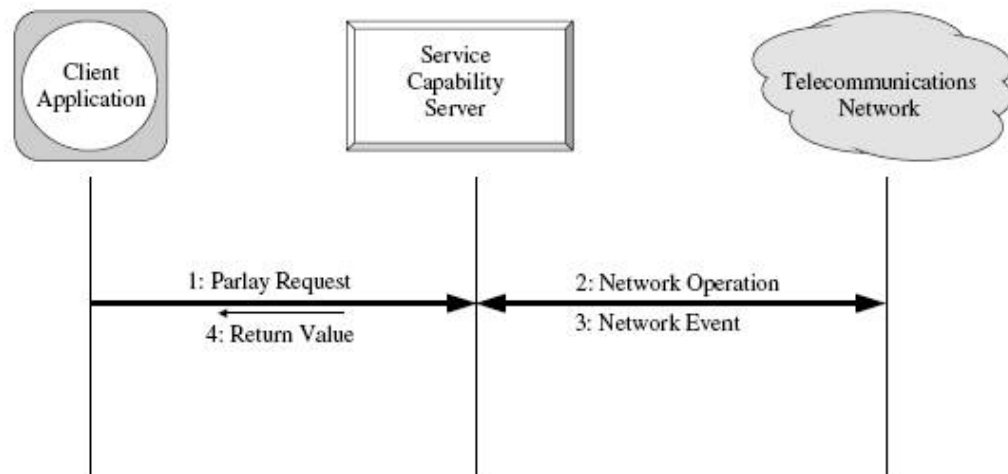
4.3.2 Métodos síncronos o asíncronos. Un método de Parlay puede ser síncrono o asíncrono. Los métodos síncronos retornan un resultado si el procesamiento del método se completa. E resultado puede ser una excepción, un valor, o ningún valor (void). Los métodos son típicamente asíncronos cuando ellos implican mensajes enviados a través de la red. Las invocaciones a métodos asíncronos algunas veces retornan en un callback en la aplicación cuando la solicitud se completa. Estos métodos se implementan de esta forma para que la aplicación no permanezca bloqueada mientras los nodos de la red responden con el resultado de la operación. Métodos asíncronos tienen el siguiente patrón de nombres. Los nombres de los métodos implementando una solicitud tienen el posfijo Req, para ser de la forma Xreq. La aplicación que provee una interfaz de callback puede ser Xres, si es un resultado exitoso o Xerr, si es un error. Las figuras 21 y 22 muestran como funcionan estos tipos de métodos.

Figura 21. Funcionamiento métodos asíncronos



Fuente: UNMEHOPA Musa, VEMURI Kumar, BENNET Andy. Parlay/OSA: From Standards to Reality. Willey 2006. p 65

Figura 22. Funcionamiento métodos síncronos



Fuente: UNMEHOPA Musa, VEMURI Kumar, BENNET Andy. Parlay/OSA: From Standards to Reality. Willey 2006. p 64

4.3.3 Aplicaciones llevadas por eventos. La mayoría de aplicaciones en Parlay son llevadas por eventos. Un evento es el reporte de un evento

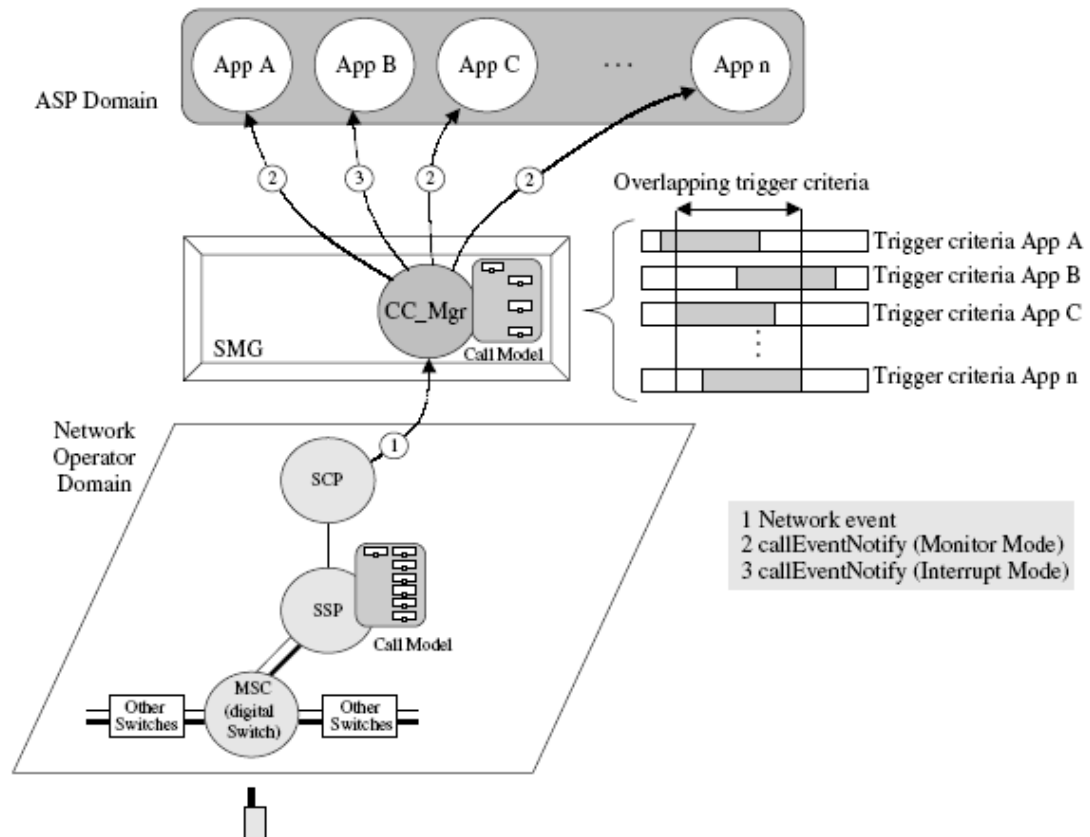
sucediendo en la red, algunos ejemplos son: llamada contestada, destino ocupado, y eventos de on-hook. Una aplicación registra su interés en determinado evento a través de una notificación. Según la implementación específica de OSA/Parlay es posible que esa notificación sea creada o registrada durante el tiempo de ejecución de la aplicación o que sea proveída por algún mecanismo offline.

Las notificaciones se encuentran generalmente compuestas por dos partes: Por un lado el alcance de la aplicación, es decir, que partes participan del evento que se está solicitando. La segunda parte hace referencia al evento como tal, es decir a sus criterios y a la acción que deba tomar la aplicación cuando se presenta el evento. La acción a tomar es definida, esto lo determina un elemento llamado MONITOR, puede tener tres modos: el primero es un modo donde no se monitorea el evento, el segundo llamado NOTIFY le notifica a la aplicación que se ha producido el evento solicitado, pero no interrumpe la llamada para entregarle control a la aplicación. El otro modo llamado INTERRUPT, interrumpe la llamada y le entrega el control a la aplicación de la llamada.

Como norma establecida en OSA/Parlay, solamente una aplicación a la vez puede tener el control de la llamada después de que se ha recibido una notificación. Es decir, si existen varias notificaciones idénticas, para varias aplicaciones, sólo una de esas aplicaciones debe tener habilitado el modo INTERRUPT del monitor. En caso de que no sea así, se arrojará la excepción P_INVALID_CRITERIA.

La figura 23 muestra como se organizan las notificaciones y de que forma no puede producirse un traslape de criterios donde el monitor esté en modo INTERRUPT.

Figura 23. Organización de notificaciones



Fuente: UNMEHOPA Musa, VEMURI Kumar, BENNET Andy. Parlay/OSA: From Standards to Reality. Wiley 2006. p 98

4.4 CAPACIDADES DE SERVICIO EN OSA/PARLAY

Como se ha mencionado anteriormente OSA/Parlay abstrae las funcionalidades de la red en una serie de Capacidades de Servicio. A continuación se explican estas capacidades, cada una de ellas compone una "parte" del estándar de OSA/Parlay, con excepción de las dos primeras partes. Junto con el nombre de la capacidad, se indican las iniciales que las identifican, estas corresponden al nombre original en inglés.

4.4.1 Parte 1 Visión General. Esta parte define todas las partes del estándar de forma general y menciona todas las definiciones y abreviaciones aplicables a las demás partes. En esta parte se explican los patrones de diseño y convenciones que se explicaron anteriormente.

4.4.2 Tipos de datos comunes. Esta parte define una serie de tipos de datos que son propios de Parlay y se encuentran implementados en Java. El uso de tipos de datos comunes ofrece ventajas de robustez y reusabilidad del código

existente, como norma todos los tipos de datos que pertenezcan a Parlay comienzan con el prefijo Tp.

4.4.3 Framework. Esta parte define la entidad conocida como Framework y las interfaces que ofrece a las aplicaciones y servicios. El Framework le provee aplicaciones mecanismos para autenticarse, descubrir servicios y acceder a ellos. El proceso de autenticación sigue los siguientes pasos, que se explicarán con detalle en la siguiente sección:

- Punto de acceso inicial
- Autenticación Mutua
- Descubrimiento de servicios
- Acceso a los servicios
- Revisión periódica de las condiciones de acuerdo de uso del servicio

4.4.4 Control de llamadas (CC). Esta parte define el SCF que reúne las funciones para realizar control de llamadas sobre la red, este control incluye llamadas multipartita y conferencias. Entre estas formas de control de llamadas existe una relación de herencia que muestra la figura 24 Esta capacidad de servicio incluye funciones para crear o habilitar la recepción de notificaciones, de esta forma la aplicación puede capturar eventos para controlar las llamadas.

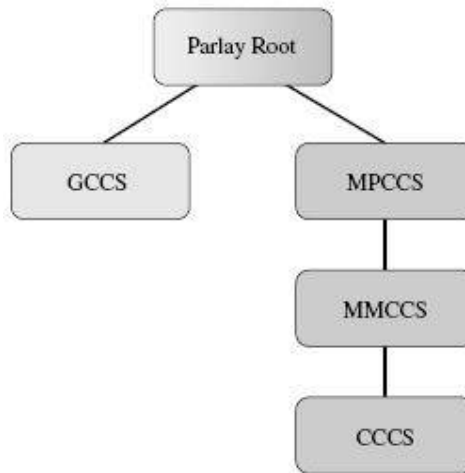
Se definen cuatro tipos fundamentales de control de llamadas:

• **Control de llamadas genérico.** Ofrece una forma simple de controlar llamadas punto a punto en la red. Esta capacidad ofrece las siguientes funciones:

- Control de llamadas simple. Está definido para controlar llamadas punto a punto
- Call gating: Esta función ofrece una técnica de control de carga utilizada por los elementos de la red para reducir el número de solicitudes que usan.
- Supervisión de llamadas. Le permite a determinadas aplicaciones realizar funciones de supervisión que pueden usarse para implementar aplicaciones como tarifas prepago.
- Enrutamiento y redireccionamiento de llamadas. Esta función permite que una llamada sea redireccionada por una aplicación.
- Consejo de cobro. Esta función permite notificarle al usuario acerca del cobro de la llamada en proceso.

- Plan de cobro. Los planes de cobro pueden ser asignados por la aplicación.

Figura 24. Relación entre los tipos de Control de Llamadas



Fuente: UNMEHOPA Musa, VEMURI Kumar, BENNET Andy. Parlay/OSA: From Standards to Reality. Willey 2006. p 94

Esta capacidad de servicio cuenta con la funcionalidad de habilitar la recepción de notificaciones que cumplan con el criterio especificado.

El control de llamadas genérico presenta ciertos determinantes, por definición solamente es posible modelar dos legs de la llamada, es decir, sólo puede usarse para llamadas punto a punto. Adicionalmente, no es posible crear llamadas desde la aplicación.

- **Control de Llamadas Multipartita.** Soporta un modelo de llamadas más complejo donde pueden presentarse legs adicionales en las llamadas, estos legs pueden ser controlados de forma independiente.
- **Control de Llamadas Multimedia.** Esta habilidad hereda de el control Multipartita, añade la función para añadir flujos de media a la llamada. Estos flujos de media pueden ser añadidos a una llamada ya existente. Es posible realizar cobros basados en el volumen de consumo.
- **Control de Llamadas de Conferencia.** Esta agrega la habilidad para establecer conferencias, reservar recursos para la conferencia tales como puentes y crear subconferencias.

4.4.5 Interacción con el Usuario (UI). Esta capacidad permite la habilidad de interactuar con el usuario, solicitarle información y enviarla hacia la

aplicación. Esta es una diferencia importante con JCC/JCAT, ya que en ese modelo, la interacción con el usuario no era una parte del núcleo de JCC/JCAT.

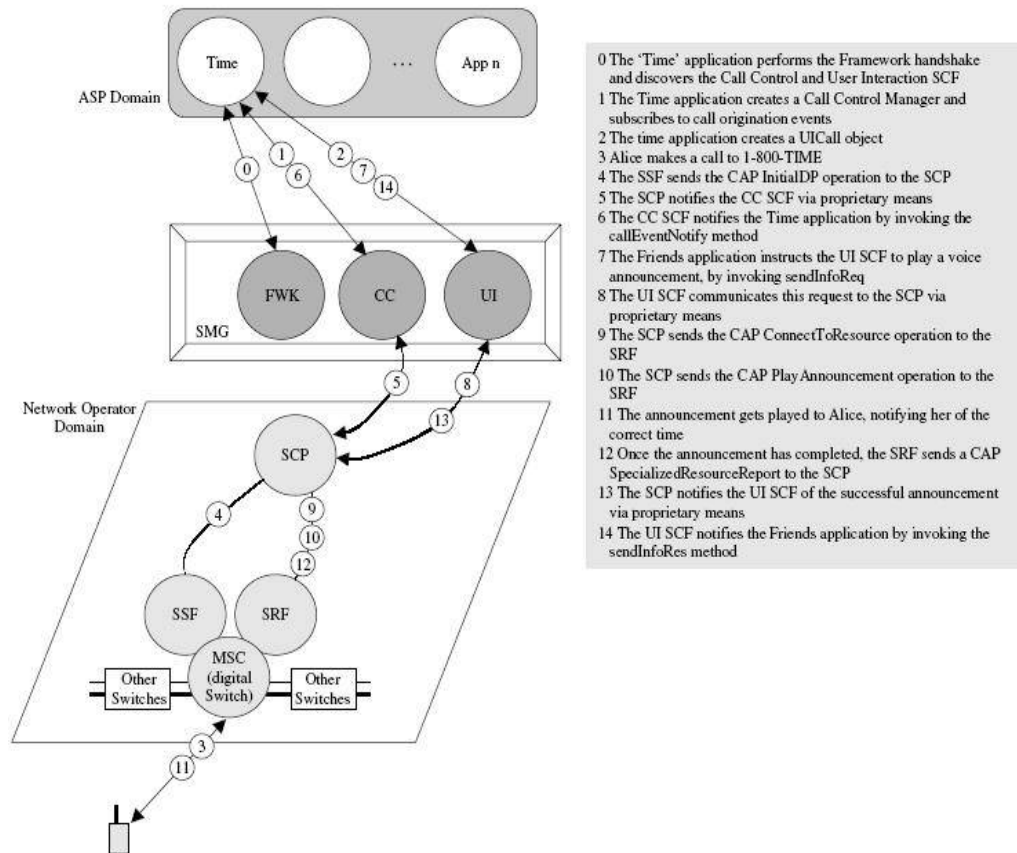
Esta capacidad incluye funciones para enviar mensajes a los usuarios y solicitar datos al usuario, por ejemplo solicitar que marque determinado número para tomar una decisión. Las capacidades de la red utilizadas para las interacciones con el usuario incluyen rangos desde grabaciones de voz y tonos DMTF en banda y SMS ó USSD fuera de banda. También puede incluir ventanas, menús o dialogos del explorador en las redes de paquetes.

Como conclusión, esta aplicación es el enlace entre los usuarios finales y las aplicaciones, es decir, permite enviar notificaciones a los usuarios y recolectar información.

Cuando se crean sesiones de interacción con los usuarios, debe enviarse una referencia a la llamada actualmente en proceso, es decir, la interacción con el usuario y el control de llamadas están estrechamente vinculados. Para pasar esta referencia del control de llamadas a interacción de usuario no se cuenta con una forma estandarizada, así que debe diseñarse una alternativa propietaria para realizar este proceso.

La figura 25 muestra un ejemplo del uso de la función llamada “interacción con usuario”.

Figura 25. Ejemplo del uso de la SCF Interacción con Usuario



Fuente: UNMEHOPA Musa, VEMURI Kumar, BENNET Andy. Parlay/OSA: From Standards to Reality. Willey 2006. p 91

4.4.6 Administración de Movilidad (MM). En las redes móviles de telecomunicaciones se añade a la complejidad existente en la red, una infraestructura que se encargue de mantener la conectividad con el usuario. No sólo a nivel de la información sino a nivel de gestión y administración del enlace, es decir, se realiza un intercambio constante de información acerca del estado de los usuarios. Esta información intercambiada siempre tiene un componente que indica la ubicación de los usuarios y a medida que cambia la ubicación de los usuarios esta información es actualizada con el fin de seguir ejerciendo un control sobre las llamadas. Como se puede apreciar, esa información puede ser utilizada por las aplicaciones.

Para esta capacidad de servicio se cuenta con un tipo adicional de comando. Hasta ahora hemos revisado los métodos síncronos y asíncronos.

Administración de movilidad implementa un tipo de comando, donde se realiza una solicitud, de tipo asíncrona de forma periódica.

Esta capacidad de servicio cuenta con más funciones que se explican más

adelante:

- **Ubicación de usuario.** Esta función permite obtener la ubicación geográfica del usuario. La forma más simple de consulta simplemente envía como parámetro las direcciones y entrega la ubicación, basada en la combinación de la latitud y la longitud, la ubicación del usuario se encuentra aproximada por la figura de un elipse ó un círculo. La solicitud avanzada le permite a la aplicación realizar una solicitud más específica.

Las regulaciones acerca de la privacidad de los usuarios es protegida a través de las regulaciones nacionales de información de cada país, ya que estas estipulan que tipo de información pueden manejar las empresas proveedores de telecomunicaciones acerca de sus suscriptores.

- **Ubicación de usuario Camel.** A diferencia de la ubicación de usuario la ubicación Camel reúne información de ubicación relacionada con la red. Esta información, por ejemplo un identificador de celular, no tiene necesariamente una relación directa con la ubicación geográfica del usuario. La información entregada por CAMEL es dependiente entonces de los identificadores que utilice cada proveedor de telecomunicaciones para identificar sus segmentos de la red.

- **Ubicación de emergencia de usuario.** Las emergencias en redes móviles son tratadas como un tipo especial de llamadas. En algunos países es una directriz que junto con la información de la llamada de emergencia la red recolecte información acerca de la ubicación de los usuarios para los casos en los cuales el usuario se encuentra inhabilitado para indicar su ubicación actual.

- **Estado de usuario.** Los dispositivos de la red mantienen información relacionada con el estado de los equipos terminales de los usuarios. De esta forma a través de este SCF las aplicaciones pueden obtener información acerca del estado de los usuarios en determinado momento. Estos estados pueden ser “no alcanzable”, “alcanzable”, “ocupado” entre otros. Estos estado se miden según la perspectiva de la red, para funciones que adicionalmente tengan en cuenta la disponibilidad del usuario debe usarse el SCF de Presencia y Administración de Disponibilidad que es explicado más adelante.

4.4.7 Capacidades de Terminal (TC). Para las redes convergentes, que posiblemente tengan que integrar redes tradicionales con redes emergentes de paquetes, resulta fundamental que las aplicaciones cuenten con la posibilidad de tener información acerca de las capacidades del terminal involucrado en determinada sesión de una aplicación.

Una aplicación con información acerca de las capacidades de terminales puede ofrecer a los equipos más avanzados las funciones más exóticas, la resolución más alta o una experiencia multimedia a sus usuarios. Mientras que puede

ofrecer las funciones esenciales y sencillas para los terminales que tengan limitadas sus capacidades.

Para obtener información acerca de las capacidades del terminal se tienen dos mecanismos: El primero consiste en una solicitud síncrona y el segundo consiste en habilitar la activación de reportes de capacidades de terminal.

Se debe tener en cuenta que la habilidad de recuperar las capacidades de un terminal está vinculada con el terminal como tal, es decir, el terminal o un nodo intermedio debe contar con la capacidad de entregar información acerca de sus capacidades.

Las capacidades retornadas son entregadas en forma de una cabecera de Perfil de Capacidades/Preferencias Compuestas (CC/PP). Esta cabecera también puede recoger las preferencias de los usuarios. Es decir, a pesar de que un equipo cuente con determinadas funciones es posible que un usuario prefiera desactivarlas o no utilizarlas.

4.4.8 Control de Sesión de Datos (DSC). Esta función permite realizar el control de sesiones donde las aplicaciones y los usuarios finales establezcan sesiones donde se intercambie información. Entre las funciones soportadas se encuentran funciones para iniciar y terminar las sesiones, para interceptarlas y funciones administrativas para tarificación.

Esto permite desarrollar aplicaciones que entreguen contenido específico a los usuarios, y que tengan las herramientas para realizar el control de ese contenido y ofrecer flexibilidad en la forma de cobro por ese contenido.

Las llamadas regulares se diferencian de las llamadas de datos en varios aspectos, entre ellos, uno de ellos es que las llamadas de datos no son orientadas a la conexión. Otras diferencias son:

- Las sesiones de datos son cobradas por la duración de la conexión mientras que las sesiones de voz son cobradas por el volumen de datos intercambiados.
- Las máquinas de estado de las llamadas y llamadas de datos son diferentes.

4.4.9 Administración de cuentas (AM). Esta SCF no se trata de un CRM, ni tampoco está encargada de manejar las cuentas con las cuales los usuarios acceden a los servicios de red como su nombre podría indicar. Esta capacidad de servicio está vinculada con la cuenta que gobierna los servicios o bienes adquiridos por el usuario a través de transacciones electrónicas realizadas a través de la red. Esta capacidad permite monitorear datos de cuentas y la disponibilidad de crédito para determinado usuario.

Los métodos invocados por esta capacidad de servicio son asíncronos, porque puede tardar un tiempo recuperar la información de todas las cuentas

necesarias.

4.4.10 Cobro basado en contenido. Con las capacidades de servicio presentadas hasta el momento se puede apreciar que las aplicaciones en Parlay están orientadas a entregar muchos tipos de contenido a sus usuarios finales. Esta capacidad de servicio vincula las funciones necesarias para realizar el cobro de ese contenido entregado. El objetivo de esta capacidad es cobrar por este contenido adicional de forma eficiente y transparente.

Esta capacidad de servicio incluye las interfaces IpChargingManager y IpAppChargingManger para habilitarle a la aplicación la creación de objetos de administración y hacer solicitudes de inicio de una sesión de cobro. Una vez que se ha creado esa sesión, las interfaces IpChargingSession y IpAppChargingSession son utilizadas para invocar eventos relacionados con el cobro.

Con este SCF existe el interés de que cuente con la capacidad de recuperarse de los posibles errores que tenga el sistema, especialmente al tener en cuenta que esta capacidad manejará transacciones monetarias. Las solicitudes cuenta con métodos para indicar si la transacción realizada fue exitosa o fallida, en caso de que se produzca un fallo, al reportarse, se reestablecen los valores anteriores a la solicitud de la transacción.

4.4.11 Administración de Políticas. Es el encargo de la administración de políticas para crear y mantener servicios de comunicación de alto nivel. La filosofía de esta SCF se basa en los principios de QoS. La aplicación de políticas de calidad del servicio resulta vital al tener la posibilidad de que terceras personas desarrollen aplicaciones, en esas circunstancias es necesario que se asuman ciertos compromisos de desempeño por parte de los desarrolladores, para que no se degrade la calidad de experiencia del usuario.

4.4.12 Administración de presencia y disponibilidad. Esta capacidad aprovecha algunos procesos de registro que distintos operadores hacen con sus terminales, entregando información acerca del estado de los suscriptores. Con esta información es posible recuperar el estado y la disponibilidad de determinados usuarios y a través de qué medios pueden ser contactados. El objetivo de este SCF es imnar la información acerca de las entidades a través de la red y publicar esa información hacia las aplicaciones. Esto incrementa la experiencia de los usuarios porque les permite que sean localizados y que su información se envíe más fácilmente. También puede evitar tráfico innecesario en la red ya que una aplicación puede decidir no enviar información hacia medios alternativos si el usuario se muestra como no disponible. La presencia se refiere al estado de un usuario como tal, es decir, si se encuentra disponible o no en determinado medio. La disponibilidad está relacionada con la configuración que establezca el usuario, por ejemplo “de todos mis colegas, sólo mi jefe puede llamarme los fines de semana”

4.5 ACCESO A LOS SERVICIOS Y LAS APLICACIONES

Como se ha explicado anteriormente, una de las características de OSA/Parlay es el esquema de seguridad que maneja para evitar a aplicaciones no autorizadas acceder a los recursos de la red ejercer control sobre las llamadas.

Es función del Framework implementar este esquema de seguridad, que se explicará posteriormente. El Framework autoriza a una lista de aplicaciones luego de que se ha establecido un acuerdo de servicio entre el desarrollador de la aplicación y el proveedor de telecomunicaciones. De esta forma, existe un mecanismo offline, para indicarle al Framework que aplicaciones deben ser autorizadas. La implementación de este mecanismo es propietaria. Los pasos que debe seguir una aplicación son los siguientes y se explicarán a continuación:

Antes de que una aplicación pueda tener acceso a un servicio, debe registrarse el servicio primero. Los pasos son los siguientes:

- Interactuar con las interfaces de autenticación del framework para autenticar mutuamente los servicios y el framework.
- Después del proceso de autenticación, el servicio le solicita acceso a las interfaces del framework para registrarse.

La aplicación para tener acceso a los servicios registrados según los pasos anteriores debe realizar los pasos a continuación:

- Interacción con la interfaz inicial del framework para autenticar mutuamente el framework y la aplicación.
- Después de la autenticación satisfactoria, la aplicación solicita acceso a las interfaces framework tales como:
 - Descubrimiento de servicio
 - Uso del servicio a través de la interacción con la administración del acuerdo de servicio.

4.5.1 Acceso a la interfaz inicial. Una aplicación comienza su sesión de Parlay invocando `initiateAuthenticationWithVersion()` en la interfaz `IpInitial` (Figura 28). Una aplicación conoce acerca de la referencia a la interfaz inicial a partir de métodos offline. El argumento de `initiateAuthenticationWithVersion()` es la versión de el Framework de Parlay., un identificador para la aplicación y su dominio. En respuesta, el Framework entrega una referencia a la interfaz de autenticación.

Cuando se está interactuando con `initiateAuthenticationWithVersion()`, la

interfaz de autenticación de la aplicación debe ser pasada. La aplicación debe también identificarse populando la unión TpDomainID.

El método `initiateAuthenticationWithVersion()` retorna una instancia del tipo `TpAuthDomain`. Esta variable incluye una referencia a la aplicación del Framework `IpApiLevelAuthentication`, que implementa los métodos de autenticación de Parlay.

Esta interfaz implementa un proceso de autenticación que es conocido como `P_OSA_AUTHENTICACION` y es solicitado por la aplicación cuando invoca `initiateAuthenticationWithVersion()` en la figura 26.

El último argumento de `initiateAuthenticationWithVersion()` es `SP_PARLAY_4`, una constante que es conocida para esta implementación del Framework conforme con la versión de Parlay 4.

4.5.2 Autenticación del Framework y la aplicación. Como se indica en la figura 28 la aplicación autentica el Framework vi la interfaz `APILevelAuthentication` que ha obtenido. Tanto la aplicación como el Framework pueden continuamente autenticarse, a través de intervalos aleatorios o por acuerdo de servicio. Si la aplicación o el Framework fallan el proceso de autenticación, la sesión de acceso y cualquier uso de sesiones es abortado.

En la figura 4 muestra un intento de reto. Después, un objeto `Challenge` es instanciado; `Challenge` es una clase de utilidad privada que formatea el reto de acuerdo a la sección 4.1 de RFC

Figura 26. Acceso a la interfaz inicial

```
1. ORB orb = ORB.init(args,null);
2. // Obtain reference to IpInitial, e.g. By reading a file
3. BufferedReader in = new BufferedReader( new FileReader(...));
4. String initialAddr= in.readLine();
5. in.close();
6. //convert string to IpInitial
7. org.omg.CORBA.Object
   obj=orb.string_to_object(initialAddr);
8. IpInitial = initial = IpInitialHeler.narrow(obj);
9. //create an object that implements IpClientAPILevelAuthentication
   ClientAPILevelAuthentication
10. ClientAPILevelAuthenticationImpl clientAuthentication = new
   ClientAPILevelAuthenticationImpl(...);
11. orb.connect(clientAuthentication);
12. //populate arguments and invoke
   initiateAuthenticationWithVersion
13. TpDomainID domainID = nre TpDomainID();
14. domainID.ClientAppId("Example.com/client");
15. TpAuthDomain
   authDomain = initial.initiateAuthenticationWithVersion(new
   TpAuthDomain(domainID, clientAuthentication),"P_OSA_AUTHENTICATION",
   "SP_PARLAY_4");
```

Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 183

El Framework retorna una respuesta utilizando el mecanismo de autorización previamente puesto y de acuerdo a la sección 4.1 La aplicación pasa el arreglo de bytes retornado a un objeto de respuesta. Se comparan los objetos de reto y respuesta. En caso de ser exitosa la comparación la aplicación invoca el método `authenticationSucceeded()`.

4.5.3 Solicitando acceso a la interfaz de descubrimiento de servicios.

Asumiendo que tanto la aplicación y el Framework han completado exitosamente el proceso de autenticación, la aplicación puede solicitar acceso a las capacidades del Framework tales como descubrimiento después de invocar `requestAccess()`.

Esto le permite a la aplicación descubrir los servicios y sus detalles son expresados como propiedades que son ofrecidos.

Para obtener una referencia a la interfaz de acceso estándar, la aplicación puede proveer una referencia para su propia interfaz. La interfaz de acceso de la aplicación soporta un método `terminateAccess()`, que puede usar el Framework si lo considera apropiado.

Figura 27. Autenticación ante el Framework

```
1. //Convert reference to FW's authentication interfaces
2. IpAppLevelAuthentication authentication =
   IpAppLevelAuthenticationHelper.narrow(authDomain.AuthInterface);
3. authentication.selectAuthenticationMechanism("P_OSA_HMAC_SHA1_96");
4. //Set challenge text
5. byte[] value = new String(...).getBytes();
6. // Class Challenge is a utility object that formats the challenge text
7. // according to section 4.1 or RFC 1994
8. Challenge challenge = new Challenge(value);
9. byte[] response = authentication.challenge(challenge.getBytes());
10. //To decode the response properties are passed to the Response class
11. Properties props = new Properties();
12. props.setProperty("hashAlgorithm", "HmacCSHA1");
13. props.setProperty("fwKey", ...);
14. props.setProperty("fwSharedSecret", ...);
15. //Class Challenge is a utility object that validates the response
16. //according to section 4.1 of RFC 1994 If validation fails at any
17. //stage an exception is thrown. In response the authentication
18. // process is aborted. Otherwise,
   authentication has succeeded.
19. Try{
20. Response re = new Response(response, challenge, props);
21. } catch (Exception e) {
22.     authentication.abortAuthentication();
23.     throw new Exception(...);
24. }
25. authentication.authenticationSucceeded();
```

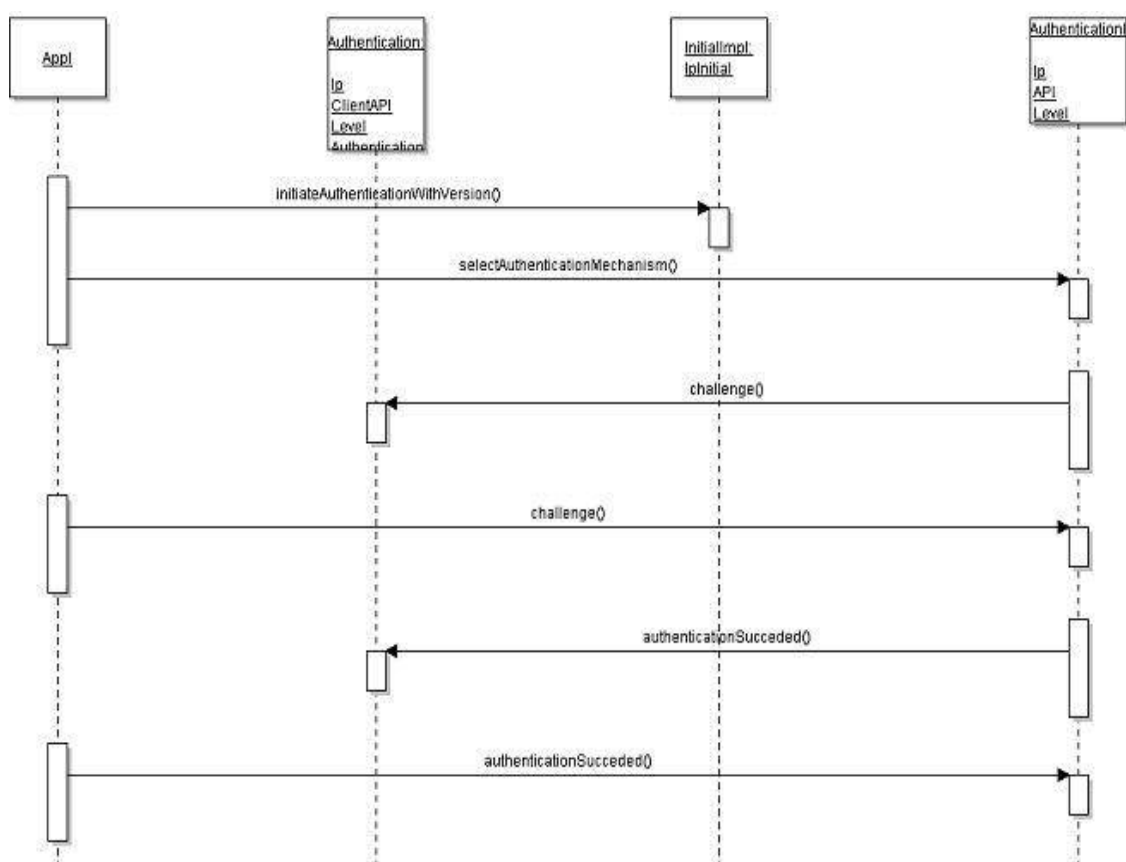
Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 184

Después de tomar y castear la interfaz de acceso a través de requestAccess() la aplicación debe invocar selectSigningAlgorithm(). Este método indica el mecanismo usado para crear y validar firmas digitales. Después la aplicación toma y castea la interfaz de descubrimiento a través de la invocación obtainInterface() con argumento P_DISCOVERY.

4.5.4 Descubriendo Servicios Parlay. La aplicación puede descubrir servicios Parlay y sus detalles utilizando la interfaz del Framework IpServiceDiscovery. (Figura 30). Esta funcionalidad es útil en el caso de que nuevos servicios sean añadidos.

La figura 31 muestra el acceso a un servicio del tipo MultiMediaControl, se especifican distintos parámetros con los valores aceptados para tal tipo de servicio.

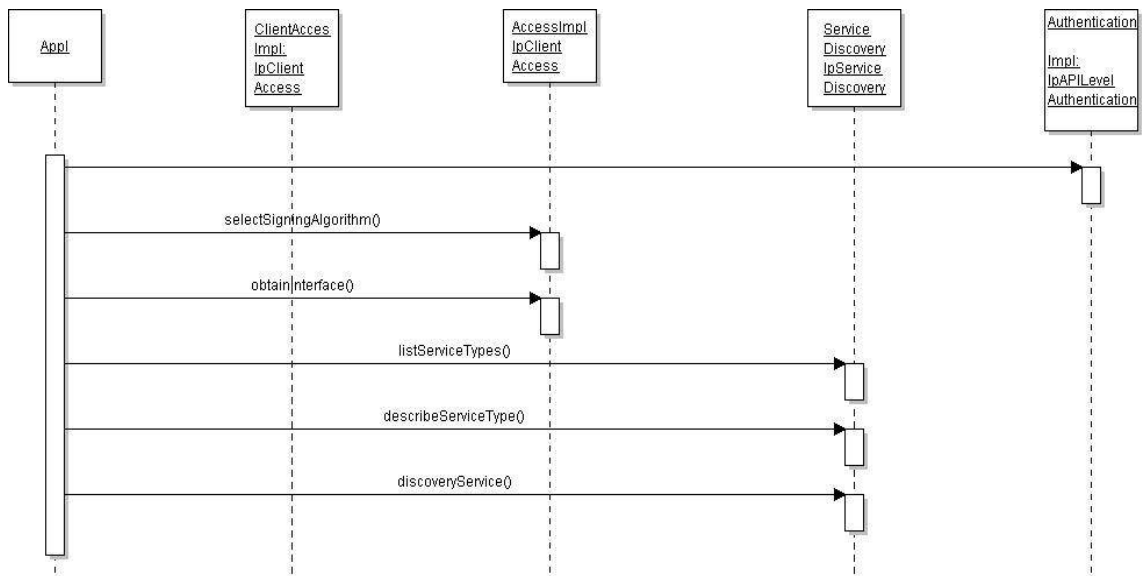
Figura 28. Diagrama de secuencia proceso de autenticación



Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 185

El diagrama de secuencia de este proceso se indica en la figura 29.

Figura 29. Descubrimiento de servicios



Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 186

Figura 30. Uso de la interfaz IpServiceDiscovery

```

1. //Create a ClientAccess object and publish it
2. ClientAccessImpl clientAccess= new ClientAccessImpl();
3. orb.connect(clientAccess);
4. // Retrieve the access interface reference and convert it
5. obj = authentication.requestAccess("\P_OSA_ACCESS", clientAccess);
6. IpAccess acces = IpAccessHelper.narrow(obj);
7. // Need to set a signing algorithm in case digital signatures
8. //are required
9. String SigningAlgorithm= acces.selectSigningAlgorithm("\P_SHA1_DSA");
10.// Retrieve the discovery interface reference and convert it
11.obj = acces.obtainInterface("\P_DISCOVERY");
12.IpServiceDiscovery serviceDiscovery = IpServiceDiscoveryHelper.narrow
    
```

Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 187

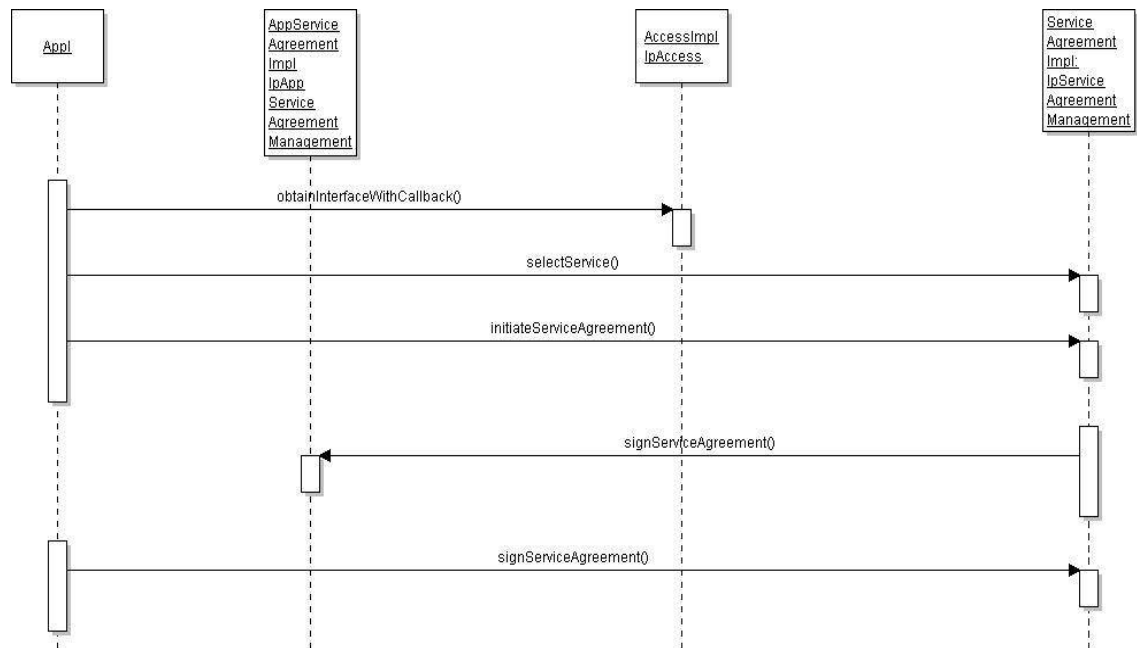
Figura 31. Descubrimiento del servicio Multimedia Call Control

```
1. //Create and publish a clientServiceAgreementManagement object.
2. IpAppServiceAgreement
   clientServiceAgreementMgmt = new AppServiceAgreementManagementImpl(this);
3. orb.connect(clientServiceAgreementMgmt);
4. //Exchange the serviceAgreement interface references and convert
5. obj=access.obtainInterfaceWithCallback("P_SERVICE_AGREEMENT_MANAGEMENT",
   clientServiceAgreementMgmt);
6. IpServiceAgreementManagement
   serviceAgreement=IpServiceAgreementManagementHelper.narrow(obj);
7. // Obtain a serviceToken based on discovered service identifier
8. String serviceToken=serviceAgreement.selectService(
   services[0].ServiceID);
9. // Tell Framework to initiate service agreement process
10. serviceAgreement.initiateSignServiceAgreement(serviceToken);
11. // wait here for Framework to complete signServiceAgreement invocation
12. // on previously passed clientServiceAgreementMgmt.
13. String agreementText = "The agreed upon text";
14. // Class CMS (Cryptographic Message Syntax) is a utility object that
15. // formats the agreement (as defined in [RFC 2630])
16. CMS cms = new CMS(agreementText);
17. TpSignatureAndServiceMgr signAndServ=serviceAgreement.signServiceAgreement
   (serviceToken, cms.getCMSText(), signingAlgorithm);
18. Properties props = new Properties();
19. //pass the CMS verifying function necessary properties as the location
20. // of the public Framework key, etc-
21. try{
22.     new CMS(signAndServ.DigitalSignature,cms.getCMSText(),props2);
23. }
24. catch (Exception e){
25. return;
26. //Finally gets access to the MultimediaCallControl service!!!
27. IpMultimediaCallControlManager mmccMgr=
   IpMultiMediaCallControlManagerHelper.narrow(signAndServ.ServiceMgrInterface);
```

Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 188

4.5.5 Firmando el acuerdo de servicio. Si la aplicación desea usar el servicio que ha descubierto, debe colaborar con el Framework para acordar los términos de uso. La interfaz para hacer esto es `IpServiceAgreementManagement` y es también tomada a través de la interfaz `IpAccess`. Esta vez, sin embargo, para garantizar que ambos lados acuerden y firmen el acuerdo, la aplicación debe proveer una interfaz de callback llamada `IpAppServiceAgreementManagement`. Esto se muestra en la figura 32.

Figura 32. Firma del acuerdo de servicio

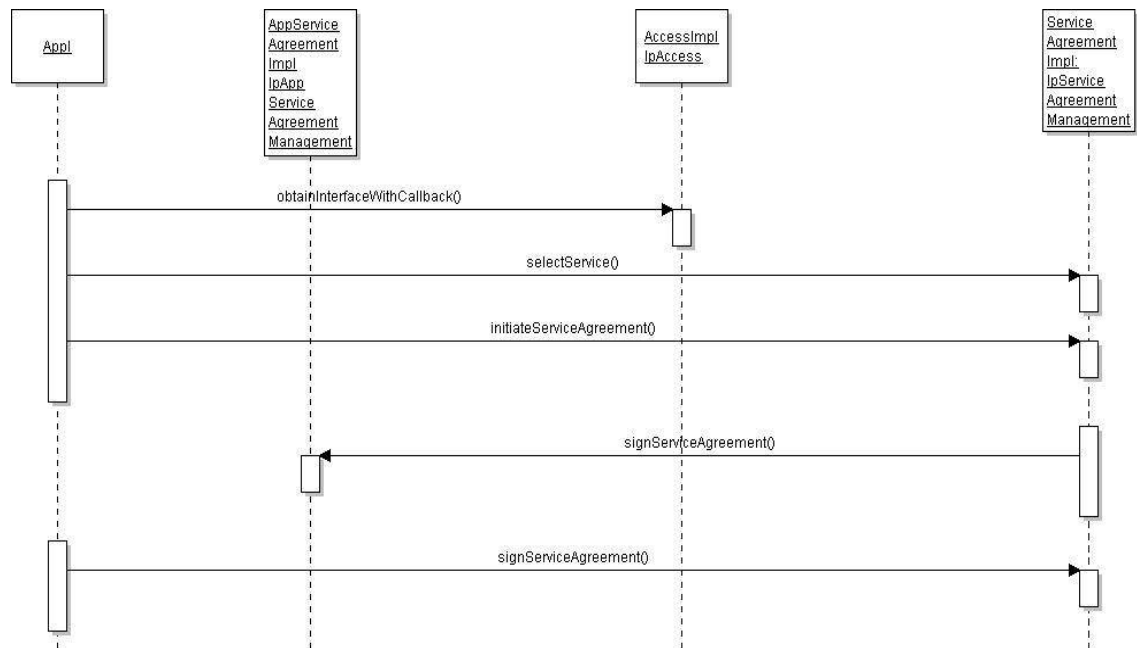


Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 190

El diagrama de secuencia de la figura 33 indica el acuerdo de servicio mutuo.

El identificador de servicio del servicio seleccionado es tomado en la fase de descubrimiento. Este identificador pasa a la interfaz de acuerdo de servicio para que el Framework pueda crear un serviceToken invocando selectService(). Ese token contiene información específica del operador de red relacionada con el acuerdo del servicio. Cuando se recibe el serviceToken, la aplicación debe preguntar al Framework para iniciar el acuerdo de servicio invocando initiateServiceAgreement(). En el proceso de acuerdo del servicio se utilizan de nuevo firmas digitales para asegurar la autenticidad. Si la verificación es exitosa, la aplicación inicia su proceso de aceptar el acuerdo de servicio. Si el Framework acepta los términos del acuerdo de servicio, retorna su firma digital junto con una referencia al servicio solicitado.

Figura 33. Acuerdo mutuo de servicio



Fuente: JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. p 189

En ese momento, la aplicación puede comenzar a utilizar las funciones del servicio adquirido haciendo uso de la referencia recibida. En resumen

4.6 MANTENIMIENTO DE LOS SERVICIOS.

Una vez que las aplicaciones han accedido a los servicios, OSA/Parlay cuenta con mecanismos para verificar el adecuado funcionamiento de los servicios, esta administración resulta necesaria debido a las posibles fallas que puede presentar una red. Existen tres tipos de controles, la administración de carga, la administración de faltas y la administración de HeartBeat.

4.6.1 Administración de Carga. Este mecanismo es encargado de indicar cuando la demanda de información supera lo que se ha acordado, o incluso de lo que están físicamente capaz de administrar. El Framework actúa como intermediario entre este intercambio de información acerca de los posibles problemas de carga. Esto puede realizarse a través de reportes enviados desde los servicios o las aplicaciones hacia el framework o a través de solicitudes de información que van hacia el Framework.

El nivel de carga representa una indicación en tiempo real de la carga de alguna entidad en particular. Este nivel solamente es reportado cuando cambia de valor. Si una aplicación desea conocer un nivel de carga de algún servicio, debe realizar una solicitud al Framework.

- **Nivel de carga.** El nivel de carga puede tomar tres posibles valores: 0, 1 y 2. 0 indica que la entidad está normalmente cargada, 1 indica que la entidad está sobrecargada y 2 indica una sobrecarga excesiva. Se determinaron estos tres valores, para que cada implementación de OSA/Parlay o incluso cada servicio indique que representa cada uno de estos valores. Es decir, para determinada implementación un valor de 0 puede representar 20 llamadas/segundo, mientras que para otra puede representar 40.

- **Estadísticas de carga.** Existe otra forma de conocer el nivel de carga de un servicio o aplicación. Es posible realizar una petición al Framework para que entregue una estadística de los niveles de carga de un intervalo de tiempo determinado. Debido a que es posible que tome cierto tiempo en compilarse la información, esta operación es asíncrona. Cuando se entregan las estadísticas se entregan dos tipos de datos, el primero corresponde al nivel de carga que se ha explicado anteriormente, el segundo corresponde con un porcentaje de carga. El significado de este porcentaje y la referencia con que se mide deben ser determinados e interpretados por la implementación de Parlay.

4.6.2 Administración de Faltas. La administración de faltas es encargada de intercambiar información acerca de los problemas que se pueden presentar entre los SCF's, las aplicaciones y el Framework. Toda la información de faltas, pasa también por el Framework. Existen varias formas de intercambiar esta información de faltas.

La primera de ellas se utiliza para indicar que alguno de los elementos presenta un problema, uno de estos problemas puede referirse a que una aplicación en particular no puede utilizar una instancia de determinado servicio autorizado, también puede referirse a problemas internos que presente la aplicación como pérdida de conectividad con una base de datos.

La segunda forma es utilizada en caso de que alguna de las entidades deje de responder. Puede usarla una aplicación en caso de que la instancia del servicio ha dejado de funcionar. En ese caso se utiliza una solicitud de "are you still working?", esto hace que el Framework le ordene al SCF realizar un auto test y reportar el resultado. Si el SCF no puede responder a este comando del Framework, se inicia el proceso para finalizar la sesión del servicio. Esta solicitud también puede realizarse desde la aplicación hacia el Framework en caso en que este presente problemas.

La tercera forma consiste en solicitar un registro de las estadísticas de falta de alguna de las entidades involucradas. En esta forma, el Framework es responsable de reunir y entregar esta estadística a la aplicación.

4.6.3 Administración de HeartBeat. Las formas anteriores de administración de carga y falta parten del principio que las aplicaciones. Es posible que en algunos casos una aplicación o un SCF ni siquiera pueda reportar sus fallos o problemas de carga. OSA/Parlay tiene otra alternativa, que consiste en que una

aplicación periódicamente solicita señales (pulsos o beats en inglés)a las otras entidades. El intervalo de tiempo en el que se envían los pulsos puede ser modificado en un momento determinado, también puede configurarse a partir de cuantos pulsos la aplicación que recibe los pulsos debe tomar acción. Esta aproximación es más ligera que la de administración de falta o cargas, porque se envían menos datos en la red, aunque se envíen periódicamente. Otra diferencia es que estos datos no necesariamente se propagan a través del Framework.

5. DISEÑO DEL SERVICIO

Como se ha explicado anteriormente, el objetivo fundamental de las redes NGN ó redes convergentes es el rápido desarrollo de nuevos servicios que agreguen valor a las redes existentes. Desde hace dos años EMCALI cuenta con una red de este tipo instalada, pero hasta el momento no se han desarrollado servicios por terceras personas que, iniciamlente validen las capacidades de la red, e incrementen el valor de las redes ofrecidas a sus usuarios. El objetivo fundamental de este proyecto es el diseño e implementación de un servicio que valide el modelo de creación de servicios propuesto por el grupo de investigación de la GITI de la Universidad Autónoma de Occidente y a la vez valide las capacidades de la red multiservicios de EMCALI. Este capítulo presenta un resumen del proceso de diseño del servicio y muestra los resultados obtenidos en su implementación.

5.1 OBJETIVO DEL SOFTWARE

El desarrollo de este servicio está orientado a cumplir dos objetivos centrales:

- Validar el modelo de desarrollo de servicios propuesto por el grupo de investigación GITI de la Universidad Autónoma de Occidente .
- Validar las capacidades de la red multiservicios de EMCALI para el desarrollo de servicios y aplicaciones por terceras personas.

Como se puede apreciar estos objetivos apuntan a la validación de un modelo teórico y las capacidades de una red real, por esa razón, el servicio propuesto consiste en un servicio sencillo, pero que implemente funciones de una de las capacidades de servicio de OSA/Parlay. Debido a que ya se habían presentado otras experiencias que no pudieron llevar el servicio a una implementación real ²⁷ se decidió que desde el cominenzo se trabajaría en el entorno real de la plataforma de aplicaciones de EMCALI.

5.2 ALCANCE

El alcance de este proyecto es la validación del modelo propuesto y la capacidad de la plataforma de aplicaciones de la NGN de EMCALI. De esta manera, el alcance espcífico de este proyecto consiste en realizar una prueba completa de una de las funciones de las capacidades de servicio de OSA/Parlay. Fue escogida la capacidad de servicio de control genérico de llamadas por ser una capacidad sencilla de manejar, con un alto nivel de madurez y que ofrece muchas posibilidades para próximos servicios y aplicaciones. El alcance de este proyecto incluye el diseño e implementación del servicio.

27 VALLECILLA p36

La implementación de este servicio resulta crucial para validar el diseño propuesto. Existen dos razones para justificar eso, por un lado para esta plataforma hay muy poca documentación disponible, lo que quiere decir que un diseño sin implementación no cuenta con suficiente fundamentación teórica. Por otro lado, no existen propuestas anteriores de diseños funcionales para replicar. De esta manera, sólo a través de la implementación y las pruebas puede verificarse la validez del diseño para que cumpla los objetivos propuestos.

5.3 DEFINICIÓN DEL SISTEMA

El sistema debe utilizar la capacidad de servicio de Control de Llamadas para realizar el redireccionamiento de un tipo de llamada en particular. El tipo de llamada estará especificado por la notificación habilitada. Las llamadas se redireccionarán a números preestablecidos y configurados de forma offline por los usuarios.

El sistema debe utilizar las herramientas brindadas por el Parlay Client Hub instalado en la plataforma de aplicaciones de EMCALI.

El elemento físico que utilizará la plataforma de aplicaciones para realizar el redireccionamiento será el Softswitch de la red, este a su vez se encargará de realizar la traducción de la orden de la aplicación a los protocolos de señalización apropiados.

Inicialmente este servicio se probará con clientes de la llamada Red Inteligente de EMCALI, en caso de que la configuración actual de la red lo permita, se extenderá a usuarios de la red tradicional telefónica de EMCALI.

5.4 LISTA DE REQUERIMIENTOS FUNCIONALES

RF_01: El servicio es llamado “Portabilidad Numérica” y consiste en un número que al ser marcado debe redireccionar la llamada al número que halla configurado previamente el usuario.

RF_02: El servicio debe procesar las notificaciones enviadas por el softswitch y que estén configuradas como notificaciones de interés.

RF_03: El servicio debe enviar órdenes al softswitch para controlar las llamadas que sean de su dominio.

RF_04: El servicio debe contar con un sistema, que permita reporte el funcionamiento del servicio.

RF_05: El servicio sólo debe enviar las órdenes necesarias al Softswitch y debe liberar la llamada cuando realice el direccionamiento.

5.5 REQUERIMIENTOS NO FUNCIONALES

RNF_01: El servicio debe implementar la funcionalidad de la NGN de EMCALI, para lograr esto, debe estar instalado en la plataforma de aplicaciones existente en la red multiservicios de EMCALI.

RNF_02: El servicio debe implementar las herramientas ofrecidas por el Parlay Client Hub de ZTE.

RNF_03: La instalación y funcionamiento de la aplicación no debe requerir la instalación o modificación de hardware adicional.

RNF_04: La aplicación sólo debe acceder a las capacidades de servicio previamente.

RNF_05: La aplicación debe contar con una herramienta de para manejar las fallas y sobrecargas que pueda presentar la aplicación.

5.5.1 Seguridad: Teniendo en cuenta que las aplicaciones desarrolladas en OSA/Parlay sobre redes NGN podrían llegar a tener control de las llamadas o los recursos de la red subyacente, resulta fundamental implementar un esquema adecuado de seguridad.

Este esquema de seguridad es aplicado a través de la intervención del Framework que se encarga de autorizar a las aplicaciones el uso de los servicios. El Framework se encarga de autenticar cada una de las aplicaciones, comparando con su propio listado de aplicaciones autorizadas y sus números de aplicación. Esta autenticación se realiza de forma periódica para incrementar la seguridad.

Para añadir aplicaciones autorizadas al Framework se debe tener acceso a la consola de mantenimiento de la plataforma de aplicaciones, esto puede realizarse de dos formas:

- La primera forma consiste en conectarse por Telnet a la plataforma, sólo los equipos pertenecientes a la red local de la plataforma pueden acceder de esta forma.
- A través de una VPN configurada adecuadamente. Sólo personal autorizado del área de Ingeniería y Servicios de EMCALI y de Desarrollo y Postventa de ZTE tienen acceso a esta VPN.

5.6 ADMINISTRACIÓN DE CARGAS Y FALTAS

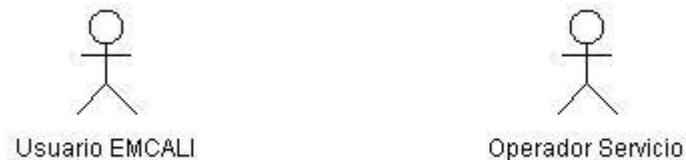
La administración de cargas y faltas es fundamental para garantizar la satisfacción de los usuarios de la red. Prever posibles fallos a corto y mediano plazo y evaluar el desempeño de la aplicación. Para este tipo de

administración, PCH exige el uso por lo menos del esquema de administración ofrecido por el HeartBeatManagement.

5.7 DEFINICIÓN DE ACTORES

La figura 34 muestra los actores que utilizarán el servicio.

Figura 34. Actores del servicio



5.7.1 Usuario de EMCALI. Suscriptores de la red Inteligente de EMCALI que acepten el servicio de Portabilidad Numérica, en caso de que la configuración de la red lo permita. Dentro de estos actores estarán incluidos los usuarios de la red tradicional telefónica.

5.7.2 Operador Servicio. Personal de EMCALI encargado de realizar el mantenimiento del servicio y atender las inquietudes de los usuarios, no se dispone de personal dedicado a un servicio en particular, por el contrario atienden las solicitudes de otros servicios ya implementados en la red.

5.8 CASOS DE USO

5.8.1 Listado de Casos de Uso

CUU 1: Redireccionamiento de la llamada.

CUA 1: Verificar funcionamiento servicio

CUA 1.1: Verificar funcionamiento

CUA 1.2: Verificar recepción de notificaciones

CUA 2: Administrar notificaciones

CUA 2.1: Crear notificaciones

CUA 2.2: Modificar notificaciones

CUA 2.2.1: Eliminar notificaciones

5.8.2 Descripción de Casos de Uso

Identificador: CUU1

Nombre: Redireccionamiento de la llamada

Actor: Usuario servicio

Descripción: Este caso de uso hace referencia a la función fundamental del servicio. Cuando el usuario que inicia la llamada marca un número, el servicio redirecciona esta llamada al número que el usuario de destino tiene configurado.

Autor: Andrés Felipe Murillo

Tabla 2. Flujo de eventos Caso de Uso Enrutamiento

Flujo de Eventos	
Curso Normal	Alternativas
1. El caso de uso inicia cuando un usuario marca a un número que cumple con los requisitos especificados en la notificación.	
2. El servicio tom control de la llamada, la suspende y busca el número al que debe redireccionar la llamada.	
3. El servicio redirecciona la llamada al número apropiado.	3.1 En caso de no haber otro número configurado, la aplicación libera la llamada, esta continua su curso habitual.
4. Al recibir una notificación de que el redireccionamiento fue exitoso, se reporta ese evento y se libera la llamada	4.1 En caso de que se reciba una notificación de enrutamiento no exitoso, se reporta ese evento y se intenta redireccionar al número original (Flujo 3.1). En caso de que el nuevo direccionamiento falle se reporta ese evento y se libera la llamada

Requerimientos especiales: N/A

Pre-condiciones: Aplicación registrada y activa
La llamada cumple con las condiciones indicadas en la notificación.

Post-condiciones: La llamada es redireccionada al nuevo número
En el reporte se agrega una operación exitosa

Identificador: CUA 1

Nombre: Verificar funcionamiento servicio

Actor: Operador servicio

Descripción: Este caso de uso hace referencia a verificar el funcionamiento planeado del servicio. Esto se puede realizar verificando la recepción de las notificaciones y el enrutamiento de llamadas.

Autor: Andrés Felipe Murillo

Tabla 3. Flujo de eventos Caso de Uso CUA1

Flujo de Eventos	
Curso Normal	Alternativas
1. El caso de uso inicia cuando el operador del servicio abre el archivo de log de la aplicación. A través de esos reportes, se puede verificar la recepción de las notificaciones y el enrutamiento de las llamadas.	

Requerimientos especiales: n/a

Pre-condiciones: Sistema de logs en funcionamiento

Post-condiciones: n/a

Identificador: CUA 1.1

Nombre: Verificar enrutamiento

Actor: Operador servicio

Descripción: Este caso de uso consiste en que el operador verifique que las llamadas sean enrutadas de la forma como se ha planeado.

Autor: Andrés Felipe Murillo

Tabla 4. Flujo de eventos Caso de Uso CUA1.1

Flujo de Eventos	
Curso Normal	Alternativas
1. El caso de uso inicia cuando el operador del servicio abre el archivo	

de log de la aplicación y verifica los reportes relativos al enrutamiento de las llamadas	
---	--

Requerimientos especiales: n/a

Pre-condiciones: Sistema de logs en funcionamiento

Post-condiciones: n/a

Identificador: CUA 1.2

Nombre: Verificar recepción notificaciones

Actor: Operador servicio

Descripción: Este caso de uso ofrece la opción de verificar que las notificaciones habilitadas sean recibidas por la aplicación.

Autor: Andrés Felipe Murillo

Tabla 5. Flujo de eventos Caso de Uso CUA1.2

Flujo de Eventos	
Curso Normal	Alternativas
1. El caso de uso inicia cuando el operador del servicio abre el archivo de log de la aplicación y verifica los reportes relativos a la recepción de notificaciones	

Requerimientos especiales: n/a

Pre-condiciones: Sistema de logs en funcionamiento

Post-condiciones: n/a

Identificador: CUA 2

Nombre: Administrar notificaciones

Actor: Operador servicio

Descripción: Este caso de uso le permite al operador administrar las notificaciones que usará la aplicación para recibir información y tomar control de las llamadas de interés.

Autor: Andrés Felipe Murillo

Tabla 6. Flujo de eventos Caso de Uso CUA2

Flujo de Eventos	
Curso Normal	Alternativas
1. El caso de uso inicia cuando se abre la consola de mantenimiento PCH-OAM y se abre el recuadro correspondiente al servicio	

Requerimientos especiales: Acceso al PCH-OAM

Pre-condiciones: Servicio registrada en PCH-OAM

Post-condiciones: n/a

Identificador: CUA 2.1

Nombre: Crear notificaciones

Actor: Operador servicio

Descripción: Este caso de uso le permite al operador crear notificaciones para que la aplicación reciba información y pueda tomar control de las llamadas de interés.

Autor: Andrés Felipe Murillo

Tabla 7. Flujo de eventos Caso de Uso CUA2.1

Flujo de Eventos	
Curso Normal	Alternativas
1. Este caso de uso inicia cuando el operador del servicio abre la consola de mantenimiento del PCH-OAM y	

abre el recuadro correspondiente al servicio.	
2. El operador hace click en la opción de crear una nueva notificación	
3. El operador configura los parámetros adecuados de la notificación.	
4. El operador almacena la notificación creada.	4.1 El operador no almacena la notificación creada. La notificación se descarta.

Requerimientos especiales: Acceso al PCH-OAM

Pre-condiciones: Servicio registrado en PCH-OAM.
PCH-OAM tiene habilitada la opción de crear notificación

Post-condiciones: El servicio recibe información sobre el tipo de llamada determinado por la notificación, puede tomar control sobre ellas.

Identificador: CUA 2.2

Nombre: Modificar notificaciones

Actor: Operador servicio

Descripción: Este caso de uso le permite al operador modificar las notificaciones para adaptar la aplicación a las nuevas condiciones que se hallan establecido.

Autor: Andrés Felipe Murillo

Tabla 8. Flujo de eventos Caso de Uso CUA2.2

Flujo de Eventos	
Curso Normal	Alternativas
1. Este caso de uso inicia cuando el operador del servicio abre la consola de mantenimiento del PCH-OAM y abre el recuadro correspondiente al servicio.	
2. El operador hace click en la notificación a modificar	

3. El operador modifica parámetros de la notificación.	
4. El operador almacena la notificación creada.	4.1 El operador no almacena la notificación creada. Los cambios realizados en la configuración de la notificación se descartan

Requerimientos especiales: Acceso al PCH-OAM

Pre-condiciones: Servicio registrado en PCH-OAM.
PCH-OAM tiene habilitada la opción de modificar notificación
Existen notificaciones creadas

Post-condiciones: El servicio puede recibir información sobre el tipo de especificado por la nueva versión de la notificación, puede capturarlas también

Identificador: CUA 2.2.1

Nombre: Eliminar notificaciones

Actor: Operador servicio

Descripción: Este caso de uso le permite al operador eliminar las notificaciones creadas para adaptar la aplicación a las nuevas condiciones que se hallan establecido.

Autor: Andrés Felipe Murillo

Tabla 9. Flujo de eventos Caso de Uso CUA2.2.1

Flujo de Eventos	
Curso Normal	Alternativas
1. Este caso de uso inicia cuando el operador del servicio abre la consola de mantenimiento del PCH-OAM y abre el recuadro correspondiente al servicio.	
2. El operador hace click en la notificación a eliminar	
3. El operador elimina la notificación	
4. El operador confirma la eliminación	4.1 El operador no confirma la eliminación de la notificación. La notificación no se elimina

Requerimientos especiales: Acceso al PCH-OAM

Pre-condiciones: Servicio registrado en PCH-OAM.

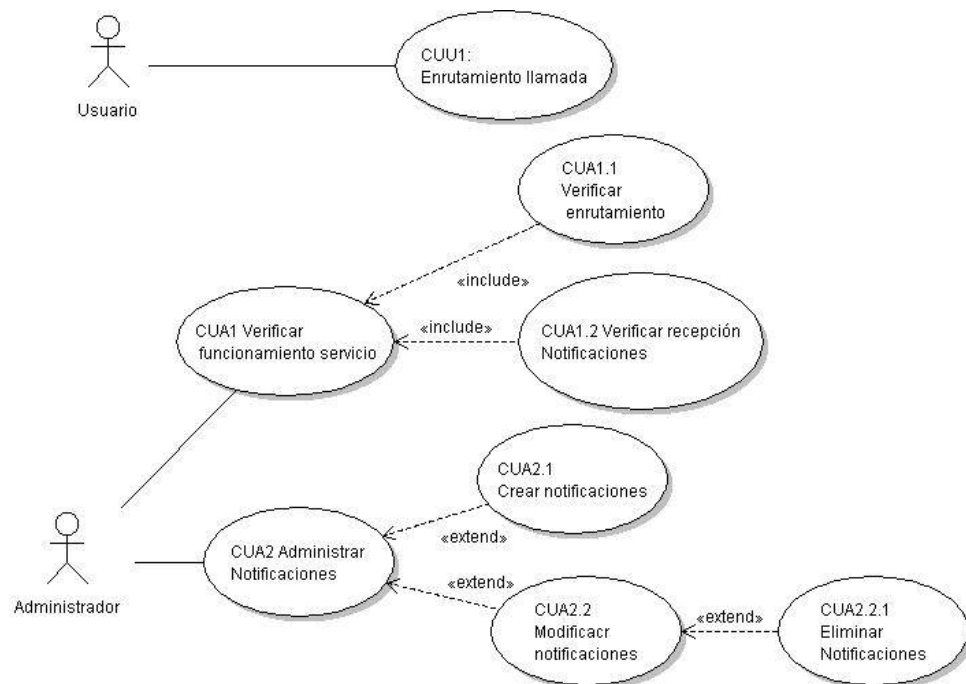
PCH-OAM tiene habilitada la opción de eliminar notificación

Existen notificaciones creadas

Post-condiciones: El servicio no recibirá información sobre el tipo de llamada especificado por la notificación eliminada. Tampoco puede controlarlas.

5.8.3 Diagramas de Casos de Uso. Las figuras 35 muestra el diagrama de Casos de Uso que reúne los casos de uso mencionados anteriormente.

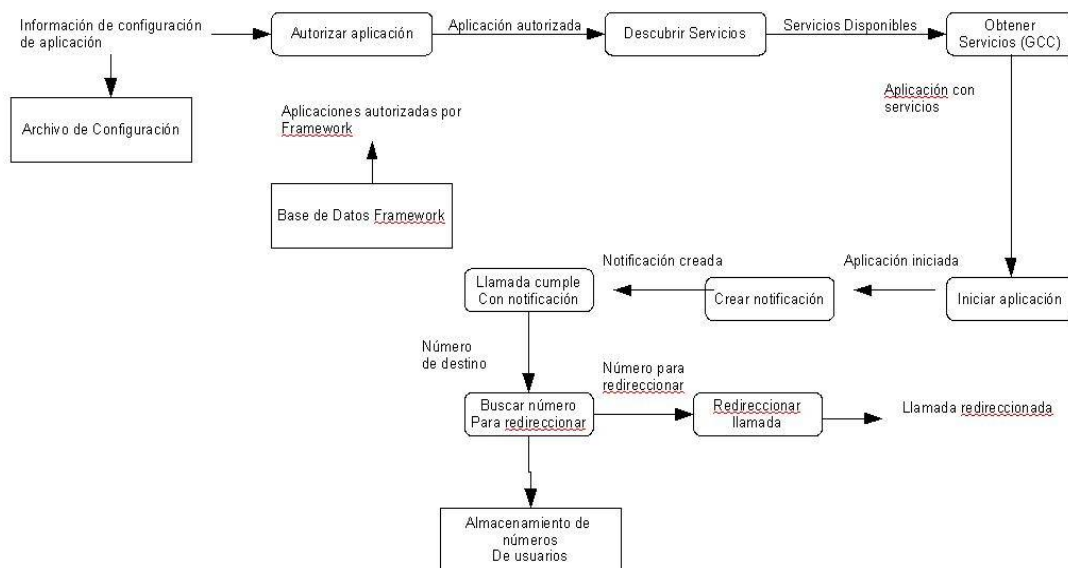
Figura 35. Diagrama de casos de uso



5.9 MODELO DE FLUJO DE DATOS

El modelo de flujo de datos que se muestra a continuación es una abstracción de alto nivel del flujo de información durante la ejecución del servicio. En los diagramas de secuencia que se mostrarán más adelante, se muestra de forma más detallada el flujo de información y las entidades involucradas.

Figura 36. Diagrama de flujo de datos servicio de Portabilidad Numérica



Como puede apreciarse en la figura 36 la aplicación maneja los siguientes datos:

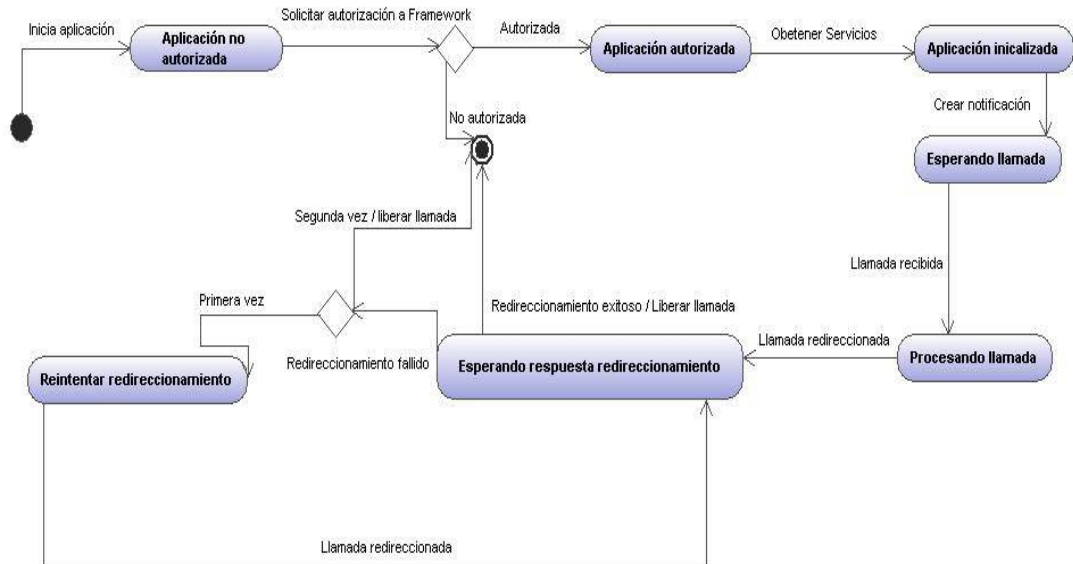
- Información de configuración de la aplicación. Estos datos se encuentran almacenados en un archivo .properties y son usados por el Framework para autorizar la aplicación y obtener información adicional sobre la aplicación. Esta información se le entrega al Framework en el momento de realizar la autorización.
- Aplicaciones y servicios autorizadas por el Framework: Los identificadores de estos se encuentran almacenados en una Base de Datos de la plataforma de aplicaciones, corresponden a acuerdos de servicio firmados previamente entre la empresa prestadora de servicios y los desarrolladores del servicio.
- Servicios disponibles: Estos servicios se encuentran almacenados en el Framework y corresponden a las capacidades de servicio de OSA/Parlay o servicios desarrollados por terceros.
- Números para direccionar: Estos números se encuentran almacenados en la memoria de la aplicación y son configurados por el usuario.

5.10 DIAGRAMA DE ESTADOS

El diagrama de estados es una abstracción de alto nivel que describe el comportamiento que tendrá la aplicación. Es decir, que estados puede tener la

aplicación y las funciones que realiza cada uno de estos estados. Este diagrama es mostrado en la figura 37.

Figura 37. Diagrama de estados



5.10.1 Explicación comportamiento de la aplicación. Al iniciarse la aplicación por primera vez esta no cuenta con la autorización para acceder a ningún servicio. En el proceso de inicialización la aplicación le entrega su información de configuración al Framework y se inicia el proceso de autenticación mutua, descubrimiento y acceso a los servicios explicado en el capítulo anterior.

Una vez la aplicación cuente con los servicios necesarios procede a habilitar la notificación con los parámetros apropiados para tomar el control sobre el tipo de llamadas especificado. Para esta aplicación las llamadas de interés son las que se dirijan hacia los números 523096X. La aplicación permanece en un estado de espera hasta que se produzca la notificación de que una llamada con ese destino se ha producido en la red.

En ese momento, el softswitch detiene la llamada y le entrega el control de la misma a la aplicación. La aplicación encuentra el número al que debe redireccionarse la llamada, usando como referencia el número de destino marcado por el usuario. Al obtenerse este número, la aplicación le ordena al softswitch redireccionar la llamada.

Como se ha explicado anteriormente, la función para redireccionar llamadas es una función asíncrona, nuevamente la aplicación ingresa en un estado de espera hasta que se reciba una respuesta satisfactoria o de error acerca del direccionamiento.

En caso de producirse un direccionamiento exitoso, la aplicación “libera” la llamada, esto quiere decir, que no recibirá notificaciones adicionales de esa llamada (esto implícitamente quiere decir que otra aplicación podría tomar control sobre ella).

En caso de producirse un error en el direccionamiento, la aplicación intenta por segunda vez realizar el redireccionamiento, en caso de fallar nuevamente, libera la llamada.

5.11 DIAGRAMA DE CLASES

Como se ha especificado anteriormente, el diseño se realizará utilizando las herramientas ofrecidas por el API de PCH, que implementa un nivel de abstracción mayor al de OSA/Parlay. Es necesario implementar PCH para trabajar en la red multiservicios de EMCALI, porque este es el API que se encuentra directamente relacionado con el hardware de la plataforma de aplicaciones.

Como norma en PCH las capacidades de servicio del control de llamadas, que son definidas como interfaces con prefijo Ip, son implementadas en clases con nombres “Xsession”. De esta forma Session, se refiere a GCC, MPSession a MPCC, MMSession a MMCC y CCSession se refiere a CCC. Estas clases mantienen la misma relación de herencia que tienen las capacidades de servicio en OSA/Parlay, esta relación se muestra en la figura 24.

Adicionalmente las interfaces de callback, cuyo prefijo es IpApp, se encuentran implementadas por las clases llamadas DummyApplication.

Según este análisis, debe establecerse una relación de uso con las clases Xsession, para invocar los métodos en Java que se traducirán en comandos para el softswitch y los servicios desarrollados deben extender de las clases DummyApplication para sobrescribir los eventos de respuesta a peticiones y otros métodos de callback.

El Framework en PCH se encuentra abstraído en la clase CgasClientHub, con la que debe establecerse una relación de uso.

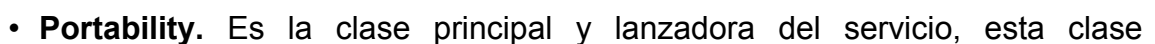
Para la elaboración del diagrama de clases y en general para la descripción de la arquitectura se utilizó el “Modelo para la creación de servicios telemáticos en la red multiservicios de EMCALI” propuesto por el grupo de investigación GITI de la Universidad Autónoma de Occidente . Este modelo, propone 3 niveles de abstracción, para este servicio se utilizó el nivel 1, que corresponde al de más bajo nivel de abstracción.

El diagrama de clases se muestra en la figura 38. Las clases desarrolladas fueron: Portability, LogResource, NotificationManager, Parameter, Utility. Las

clases Session, DummyApplication

En el diagrama de clases se muestra se verifican las relaciones de uso y herencia que se explicaron. La case principal y lanzadora del servicio es la clase Portability, utiliza la instancia que obtiene de CgasClientHub para la autenticación y acceso a los servicios. La instancia obtenida de Session para invocar los comandos de la capacidad de servicio de control genérico de llamadas y hereda de la clase DummyApplication para implementar los métodos de callback que ofrece el GCC.

5.11.1 Descripción de clases.



implementa la funcionalidad de OSA/Parlay.

Esta clase se ejecuta como un hilo, por esta razón implementa la interfaz Runnable. La ejecución de la clase como hilo tiene su fundamentación en el desempeño de la aplicación. PCH implementa herramientas para mejorar el desempeño de la clase y al iniciarse el hilo es administrada por una herramienta llamada ThreadPool que se encuentra implementada en PCH.

- **Utility.** Esta clase es utilizada por NotificationManager para habilitar el tipo de notificación necesario. Es una clase de propósito general que contiene métodos para crear los objetos de las notificaciones para los SCF's GCC y MPCC.

- **LogResource.** Esta clase es una clase que utiliza el log4j para Java. Contiene un método para que las demás clases escriban en el log los eventos o errores que se produzcan. El archivo de log es configurado en otro archivo llamado log4j.properties, tal y como se indicó en [GUEVARA].

- **NotificationManager.** Esta clase es la encargada de habilitar las notificaciones que se soliciten. Para crear los objetos de las notificaciones utiliza la clase Utility.

- **Parameter.** Esta clase es encargada de leer los parámetros almacenados en los archivos de configuración de la aplicación. Estos archivos son el archivo de configuración general de la aplicación y el archivo de configuración de la notificación.

5.11.2 Diagrama de secuencia. El diagrama de secuencia se encuentra basado en el diagrama de estados mostrado anteriormente, sin embargo este diagrama ofrece un nivel de detalle más elevado e indica las entidades involucradas en el funcionamiento del servicio.

La figura 39 muestra el diagrama de secuencia.

Al iniciarse la aplicación la primera función que realiza, durante su etapa de inicialización, es acceder al Framework para solicitar acceso a los servicios,



esto se realiza a través de la clase CgasClientHub. Una vez se obtienen los servicios la aplicación carga de su archivo de configuración parámetros relevantes e información de la notificación a habilitar. En ese momento se invoca el método run() del hilo. Posteriormente la aplicación habilita la notificación y pasa a un estado de espera.

A partir de ese momento, las llamadas que cumplan con los criterios especificados en la notificación serán interrumpidas, y se le enviará una notificación a la aplicación. Esa notificación ocasiona que el Parlay Gateway invoque el método callEventNotificationInfo(), este método de callback entrega un parámetro que tiene información sobre la llamada actual, que se encuentra suspendida. En ese momento la aplicación tiene control sobre la llamada.

Utilizando la información del parámetro entregado la aplicación busca el número al que debe direccionarse la llamada, debido a que el interés de EMCALI es validar las capacidades de su plataforma este proceso de búsqueda no se implementó a través de una base de datos, sino a través del archivo de configuración de la notificación. Sin embargo, para próximas aplicaciones es recomendado que se instale una base de datos propia para facilitar y centralizar esta información.

Cuando se obtiene el número a direccionar se invoca el método routeReq(), este método se traducirá en una orden que hará que el softswitch redirija la llamada al número especificado. Adicional a eso, cuando se invoca el método se llena un vector de solicitud de eventos, estos eventos los usa el Parlay Gateway para saber cuando invocar el método routeRes(), es decir, cuando se produzcan estos eventos se considerará exitoso el enrutamiento.

En caso de que se produzca un enrutamiento exitoso, se invoca el método routeRes(), donde la aplicación invoca el método deassignCall(), este método le indica al softswitch que la aplicación no está interesada en la llamada y libera los recursos que la aplicación esté usando.

En caso de que se presente un error en el enrutamiento, se invoca el método routeErr() y la aplicación intentará enrutar nuevamente. En caso de que se vuelva a producir un error de enrutamiento, la aplicación invocará el método deassignCall(). En ese caso la llamada se direccionaría al número original.

6. IMPLEMENTACIÓN DEL SERVICIO

La implementación del servicio se desarrolló sobre la plataforma de aplicaciones de la red multiservicios de EMCALI. Sin embargo EMCALI no cuenta con suficiente documentación para agilizar el proceso de desarrollo. Por esta razón para llevar a cabo el diseño presentado anteriormente y la implementación que se explicará a continuación fue necesario un proceso de ensayo y error, así como un análisis de caja negra de las herramientas existentes. Este proceso dificultó el proceso de diseño y dilató en gran medida los tiempos de entrega. Para finalizar la implementación del servicio es necesario utilizar todas las funciones de la aplicación PCH-OAM, estas funciones se encuentran inhabilitadas en la plataforma de servicios de EMCALI. En el momento de redacción de este informe EMCALI ha realizado una solicitud a ZTE para habilitar todas las opciones de esta aplicación. Sin embargo, no se ha obtenido respuesta por parte de ZTE, las actas de las reuniones en que se realizó la solicitud se encuentran en los anexos.

6.1 HERRAMIENTAS UTILIZADAS PARA LA IMPLEMENTACIÓN

Todo el código del servicio fue implementado utilizando Java. Esto está fundamentado en que OSA/Parlay se encuentra implementado en este lenguaje y de igual forma se encuentra implementado PCH en la plataforma de aplicaciones de EMCALI.

6.1.1 IDE. El entorno de desarrollo utilizado para este servicio fue el IBM Eclipse en su versión 3.4.1 Se utilizó este entorno de desarrollo por ser un entorno ligero, pero con capacidad de ampliar su funcionalidad a través de la adición de plugins. Adicionalmente este IDE cuenta con herramientas para crear documentación, visualizar la jerarquía de clases, los atributos de las mismas. Adicionalmente, la creación de proyectos es sencilla y los proyectos creados son fácilmente portables de un equipo a otro.

Para instalar Eclipse no se necesitan permisos de administrador, basta con contar con permisos para escritura en el disco y descomprimir la carpeta que contiene el IDE.

Eclipse se encuentra licenciado bajo la Licencia Pública Eclipse, que es aprobada como por la iniciativa de Open Source y la Fundación de Software Libre

6.1.2 PCH-OAM. Esta es la consola de mantenimiento y administración del Parlay Client Hub de ZTE. Con esta aplicación se definen y configuran los servicios y aplicaciones autorizados para el Framework, esta herramienta es esencial para el desarrollo de nuevos servicios y aplicaciones. Actualmente EMCALI cuenta con una versión de esta consola que no tiene habilitadas todas las opciones de configuración, por esta razón, no fue posible concluir todas las

pruebas de la implementación del servicio. En el momento de redacción de este informe está en proceso una solicitud por la versión completa y habilitada de esta consola.

6.2 PLAN DE PRUEBAS

Teniendo en cuenta que este servicio diseñado e implementado tiene el objetivo de validar el modelo teórico y las capacidades de la plataforma de aplicaciones y que adicionalmente se contó con escasa documentación para el proceso de desarrollo. Desde el comienzo del desarrollo se mantuvo como criterio de diseño probar los componentes desde sus versiones tempranas, esto con el fin de prever posibles causas de error en el futuro y a la vez para comprender el estado de la plataforma de aplicaciones. Como norma general, a medida que se avanzaba en el diseño se fueron probando los componentes de forma iterativa.

6.2.1 Pruebas a realizar.

Tabla 10: Prueba 1. Ejecución del código

Descripción	Ejecutar la primera versión obtenida del código
Componente validado	Máquina virtual de Java
Resultado esperado	Código empieza a ejecutarse correctamente

Tabla 11. Prueba 2. Sistema de logs

Descripción	Verificar que los eventos reportados por el Parlay Gateway y la aplicación se escriban sobre el archivo de log o se muestren en la consola de comandos
Componente validado	Sistema de logs
Resultado esperado	Archivo de logs y consola de comandos muestran eventos reportados por Parlay Gateway y aplicación

Tabla 12. Prueba 3. Inicialización de CgasClientHub

Descripción	La versión de código en prueba invoca la inicialización del CgasClientHub
Componente validado	CgasClientHub

Resultado esperado	Archivo de log reporta inicialización correcta del CgasClientHub
--------------------	--

Tabla 13. Prueba 4 Registro de aplicaciones en OAM

Descripción	Realizar el registro del servicio en el PCH-OAM
Componente validado	Registro de aplicaciones en PCH-OAM
Resultado esperado	Aplicación registrada

Tabla 14. Prueba 5. Autorización y acceso a servicios

Descripción	El servicio es autorizado por el Framework y obtiene los servicios solicitados
Componente validado	Activación de aplicaciones en PCH-OAM Añadir servicios en el PCH-OAM
Resultado esperado	Archivo de log indica que la aplicación ha obtenido los archivos solicitados.

Tabla 15. Prueba 6. Creación de notificación

Descripción	Se crea una notificación para que el servicio pueda capturar y controlar llamadas
Componente validado	Creación de notificaciones utilizando el PCH-OAM
Resultado esperado	Notificación creada exitosamente

Tabla 16. Prueba 7. Recepción de notificación

Descripción	Al establecer una llamada que cumple con los criterios de la notificación, la aplicación toma control de la llamada.
Componente validado	Comunicación entre Plataforma de aplicaciones y Softswitch
Resultado esperado	Las notificaciones son reportadas adecuadamente a la aplicación

Tabla 17. Prueba 8. Direccionamiento

Descripción	La versión de código en prueba intenta un direccionamiento
Componente validado	Capacidad de direccionamiento del código
Resultado esperado	La llamada es direccionada al número deseado

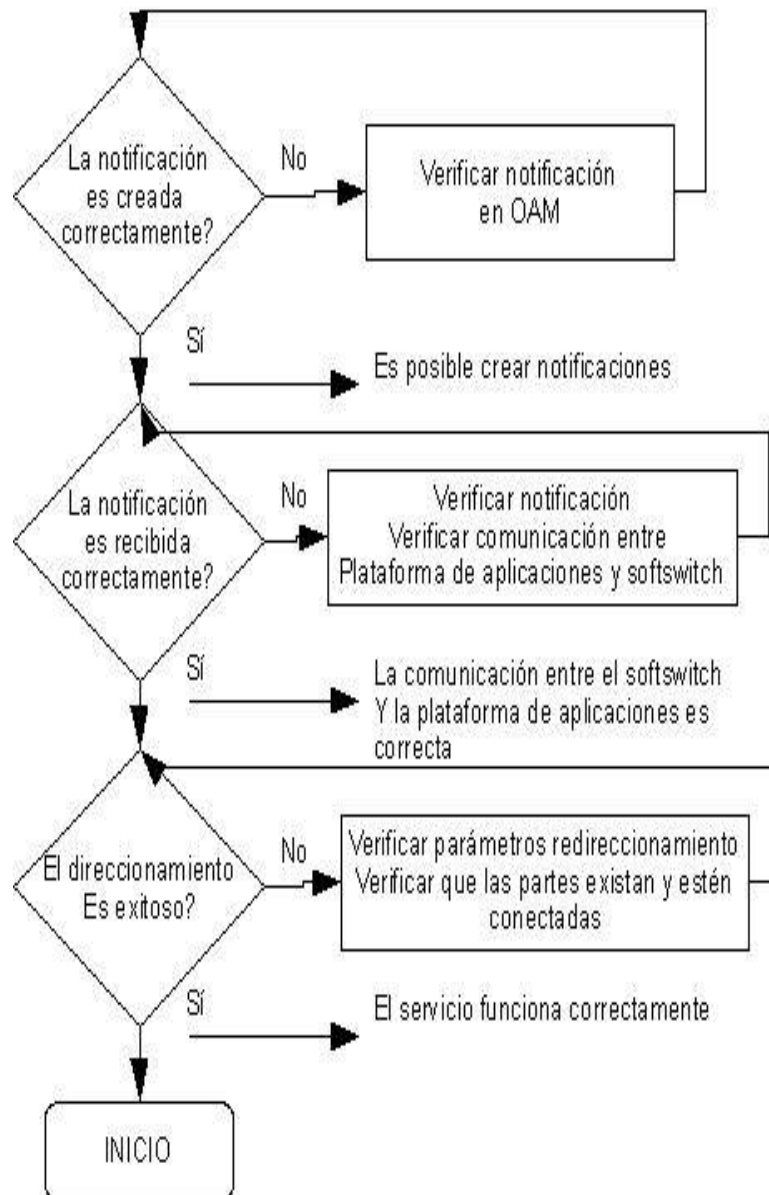
Las figuras 40 y 41 muestran los diagramas de flujo que con las pruebas para el servicio.

La figura 40 muestra las pruebas que se cumplieron a satisfacción, la figura 41 muestra las pruebas que quedaron pendientes por la falta de una versión completa y habilitada de la consola de mantenimiento OAM. Al lado de cada prueba se menciona el resultado obtenido en caso de que la prueba sea exitosa.

Figura 40. Diagrama de flujo de pruebas realizadas

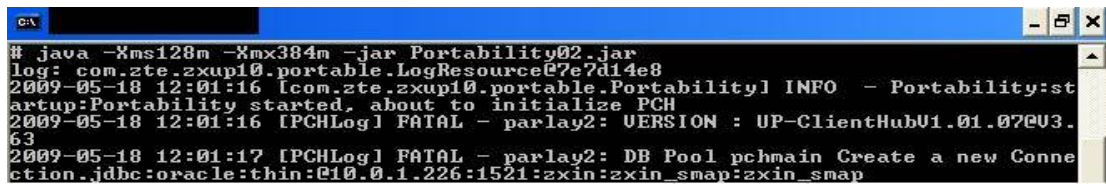


Figura 41. Pruebas sin realizar Resultados de las pruebas



- **Prueba 1.** La figura 42 muestra el momento en que la aplicación inicia su ejecución. Para que la ejecución sea exitosa es preferible especificar la cantidad de memoria mínima y máxima que utilizará la aplicación.

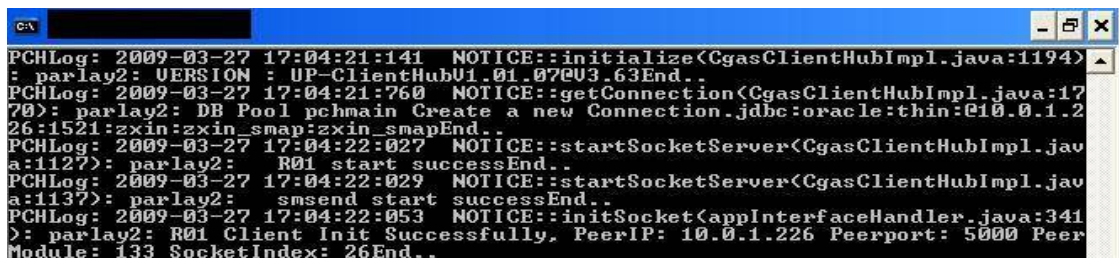
Figura 42. Ejecución de la aplicación



```
C:\> java -Xms128m -Xmx384m -jar Portability02.jar
log: com.zte.zxup10.portable.LogResource@7c7d14e8
2009-05-18 12:01:16 [com.zte.zxup10.portable.Portability] INFO - Portability:st
artup:Portability started, about to initialize PCH
2009-05-18 12:01:16 [PCHLog] FATAL - parlay2: VERSION : UP-ClientHubV1.01.070U3.
63
2009-05-18 12:01:17 [PCHLog] FATAL - parlay2: DB Pool pchmain Create a new Conne
ction.jdbc:oracle:thin:@10.0.1.226:1521:zxin:zxin_smap:zxin_smap
```

- **Prueba 2.** La figura 43 muestra una imagen del archivo de log sobre el cual la aplicación reporta los eventos a medida que la aplicación se ejecuta.

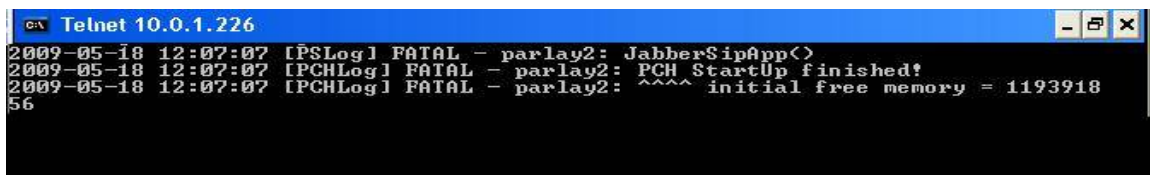
Figura 43. Funcionamiento sistema de logs



```
C:\>
PCHLog: 2009-03-27 17:04:21:141 NOTICE::initialize<CgasClientHubImpl.java:1194>
: parlay2: VERSION : UP-ClientHubV1.01.070U3.63End..
PCHLog: 2009-03-27 17:04:21:760 NOTICE::getConnection<CgasClientHubImpl.java:17
70>: parlay2: DB Pool pchmain Create a new Connection.jdbc:oracle:thin:@10.0.1.2
26:1521:zxin:zxin_smap:zxin_smapEnd..
PCHLog: 2009-03-27 17:04:22:027 NOTICE::startSocketServer<CgasClientHubImpl.jav
a:1127>: parlay2: R01 start successEnd..
PCHLog: 2009-03-27 17:04:22:029 NOTICE::startSocketServer<CgasClientHubImpl.jav
a:1137>: parlay2: smsend start successEnd..
PCHLog: 2009-03-27 17:04:22:053 NOTICE::initSocket<appInterfaceHandler.java:341
>: parlay2: R01 Client Init Successfully, PeerIP: 10.0.1.226 Peerport: 5000 Peer
Module: 133 SocketIndex: 26End..
```

- **Prueba 3.** La figura 44 es una imagen del archivo de log, el mensaje “PCH Startup finished!” indica que la inicialización de CgasClientHub ha sido exitosa.

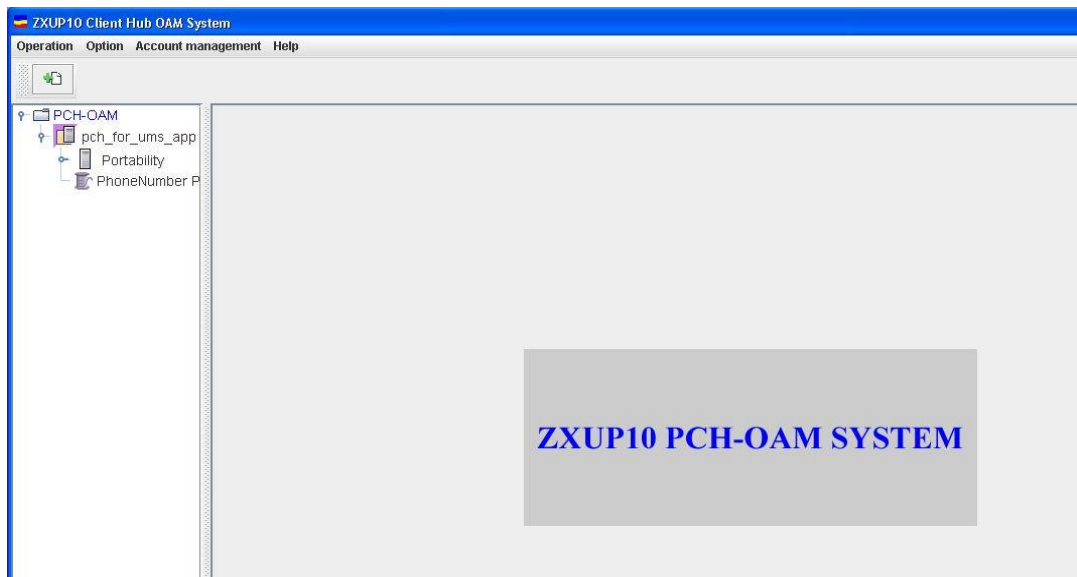
Figura 44. Inicialización exitosa de CgasClientHub



```
C:\> Telnet 10.0.1.226
2009-05-18 12:07:07 [PCHLog] FATAL - parlay2: JabberSipApp()
2009-05-18 12:07:07 [PCHLog] FATAL - parlay2: PCH StartUp finished!
2009-05-18 12:07:07 [PCHLog] FATAL - parlay2: ^^^^ initial free memory = 1193918
56
```

- **Prueba 4.** La figura 45 es una imagen tomada de la consola PCH-OAM, donde puede observarse a la aplicación registrada, no se muestran detalles por ser información propietaria de ZTE.

Figura 45 Registro de aplicación



- **Prueba 5.** En la figura 46 el mensaje P_GENERIC_CALL_CONTROL indica que la aplicación ha recibido una instancia del servicio GCC, este es el último paso en el proceso de autenticación, descubrimiento y acceso a los servicios. A partir de ese momento la aplicación puede utilizar las funciones de esta SCF.

Figura 46. Acceso a servicio GCC

```
2009-05-18 12:07:06 [PCHSCSMLog] FATAL - parlay2: Portability::P_GENERIC_CALL_CO
NTROL
@@@ Service Available @@@ P_GENERIC_CALL_CONTROL2009-05-18 12:07:06 [PCHSCSMLog]
FATAL - parlay2: &&&&&&&activateApp::appName =Portability,serviceKey = 0,Active
Flag = 1,beatLostNum = 0
2009-05-18 12:07:06 [com.zte.zxup10.portable.Portability] WARN - Portability:ini
tialize:About to load parameters from config file
```

- **Prueba 6.** La figura 47 muestra que la opción de creación de notificación no se encuentra habilitada con la versión de PCH-OAM que cuenta actualmente EMCALI. Por esa razón esta prueba y las siguientes no pudieron realizarse adecuadamente. Como se explicó anteriormente el servicio requiere que se cree una notificación para capturar un tipo determinado de llamadas y realizar el redireccionamiento, sin la creación de esta notificación el Softswitch no activará la aplicación y no puede completarse el funcionamiento del servicio. En el momento de redacción de este informe está en proceso la solicitud a ZTE de una versión del PCH-OAM que tenga habilitada esta opción para finalizar las pruebas.

Figura 47. OAM con opciones no habilitadas



7. RECOMENDACIONES AL MODELO ARQUITECTÓNICO PARA LA IMPLEMENTACIÓN DE NUEVOS SERVICIOS TELEMÁTICOS SOBRE LA RED MULTI-SERVICIOS

Como se ha mencionado anteriormente este servicio fue desarrollado para colaborar en el proceso de desarrollo y validación del “Modelo arquitectónico para la implementación de nuevos servicios telemáticos sobre la red multi-servicios” realizado por el GITI de la Universidad Autónoma de Occidente. Una vez fue finalizada la primera iteración en el desarrollo de este servicio, se recogen las experiencias de esa iteración y se presentan recomendaciones a ese modelo para ajustarlo a las condiciones reales de la red multiservicios de EMCALI. De esta forma, el modelo se construye apoyado en experiencias prácticas de la implementación de un servicio. Este capítulo presenta las recomendaciones que se realizaron a ese modelo. El modelo original se encuentra en los anexos.

Este modelo presenta tres niveles de abstracción para desarrollar servicios. Cada uno de esos niveles ofrece determinado nivel de abstracción y ofrece distintas posibilidades. Según el nivel de experiencia en OSA/Parlay el desarrollador puede escoger entre estos tres niveles. El nivel 3 es el nivel de más alta abstracción y el nivel que menor conocimiento de OSA/Parlay requiere. Igualmente es el nivel que menos posibilidades ofrece. El nivel 1 es el nivel de más baja abstracción y requiere un amplio conocimiento de OSA/Parlay y PCH, pero a la vez ofrece un rango de posibilidades más amplio.

Se realizó un análisis de cada uno de los niveles y con base en ese análisis se presentaron recomendaciones para cada nivel.

7.1 RECOMENDACIONES REALIZADAS AL NIVEL TRES

7.1.1 Capacidad de invocación de comandos. (CommandManager). La capacidad de invocación de comandos es una capacidad genérica y necesaria para el desarrollo de aplicaciones sobre NGN. Esta capacidad le permite a la aplicación ejercer cierto control sobre las llamadas así como solicitar el estado de diversos elementos de la red o enviar informes a los usuarios. Sin embargo, se debe contextualizar esta capacidad de envío de comandos. Es decir, los comandos que pueda invocar una aplicación deben depender del estado de la aplicación (autenticada, solicitando servicios, con servicios) y de los servicios adquiridos por la misma. De esta forma, se mantiene la filosofía de seguridad propuesta originalmente por los desarrolladores de OSA/Parlay.

Igualmente, la capacidad de enviar comandos debe tener algún mecanismo de respuesta para que la aplicación obtenga un retorno de los comandos realizados y respuestas en el caso de métodos asíncronos.

El comando obtener comandos, será de gran ayuda entonces para conocer los

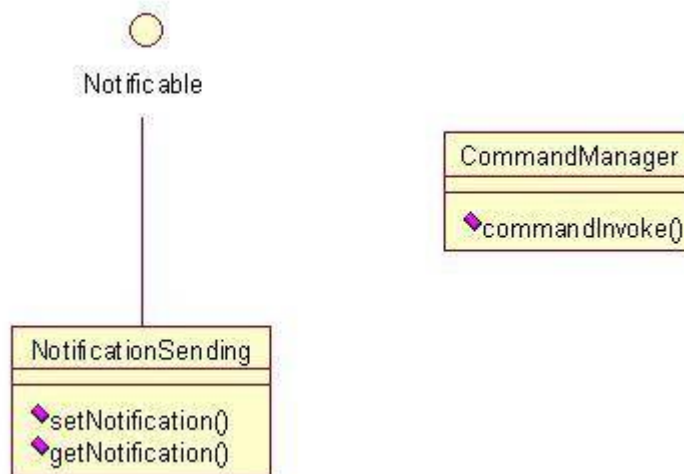
comandos disponibles según el contexto actual.

7.1.2 Envío de notificaciones (NotificationSending). El envío de notificaciones posibilita que el GW le envíe notificaciones acerca de eventos relevantes a la aplicación. A través de esta capacidad pueden implementarse servicios orientados a eventos.

7.1.3 Envío de eventos. Después de examinar el desarrollo de una aplicación de prueba, se verificó que no es necesario implementar una capacidad para el envío de eventos. Un evento es un mensaje que envía un dispositivo de red, (en el caso de NGN, traducido a SIP) que se envía al Gateway, a la vez, este lo traduce en una notificación para entregar a la aplicación información más completa y sencilla de interpretar. Como tal, las aplicaciones no manejarán directamente eventos así que se hace innecesario incluir esta capacidad en el modelo.

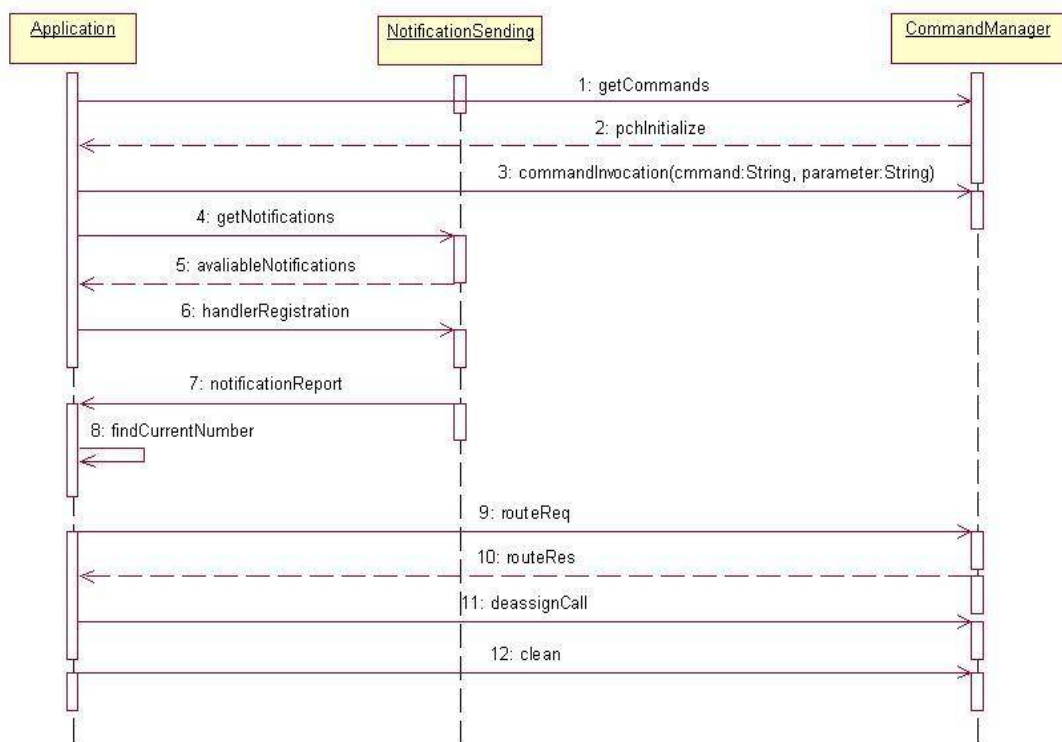
7.1.4 Diagrama de clases. El diagrama de clases de este nivel es el siguiente:

Figura 48. Diagrama de clases Nivel 3.



7.1.5 Diagrama de secuencia. Como se mencionó anteriormente, para validar que el modelo cuenta con las herramientas necesarias para desarrollar servicios en cada uno de sus niveles, se mostrará una implementación sencilla de la aplicación de prueba. Esta implementación se mostrará a través de un diagrama de secuencia.

Figura 49. Diagrama de secuencia Nivel 3. Diseño servicio de prueba



7.2 RECOMENDACIONES REALIZADAS AL NIVEL DOS

7.2.1 Control de información (InformationManager). Esta capacidad fue ampliada para permitir mostrar la información en dos direcciones, es decir, con esta capacidad es posible enviar información relevante hacia el Parlay Gateway. También puede enviar notificaciones desde el GW hacia las aplicaciones, esto se implementó extendiendo las funciones de NotificationSending del nivel 3.

7.2.2 Control de información de usuario (UserInfoManager). El nombre de esta capacidad se modificó a Control de información de usuario, ya que a través de esta capacidad también se puede enviar información hacia los usuarios y no sólo obtener información de ellos o sus terminales.

7.2.3 Control de Conexión (ConnectionManager). Es a través de esta capacidad que las aplicaciones pueden ejercer control sobre las llamadas.

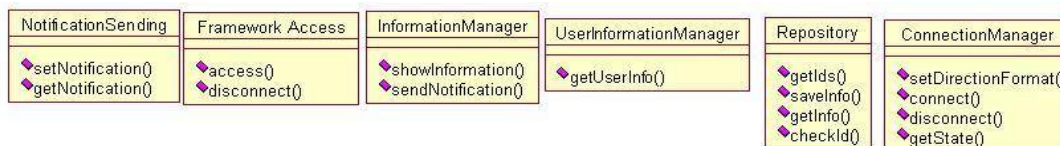
7.2.4 Repositorio (Repository). Permite realizar consultas o actualizaciones a bases de datos u otros sitios de almacenamiento de información.

7.2.5 Plataforma (Framework). En el metamodelo analizado, no se encontró una herramienta clara que permita realizar el proceso de autenticación mutua, descubrimiento de servicios y firma de acuerdo de uso del servicio. Debido a la

importancia de esta capacidad y a que requiere un nivel de seguridad adecuado, hemos añadido esta capacidad al nivel 2 del modelo. Esta capacidad implementa las funciones del Framework de OSA/Parlay

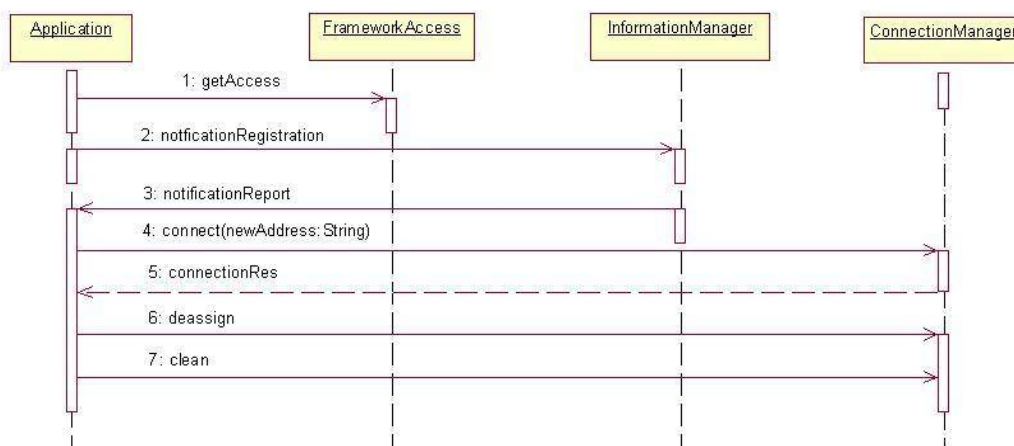
7.2.6 Diagrama de clases. El diagrama de clases de este nivel es el siguiente:

Figura 50. Diagrama de clases Nivel 2.



7.2.7 Diagrama de secuencia. El diagrama de secuencia para la implementación de la aplicación de prueba, usando este nivel, se muestra a continuación

Figura 51. Diagrama de secuencia Nivel 2. Diseño de servicio de prueba

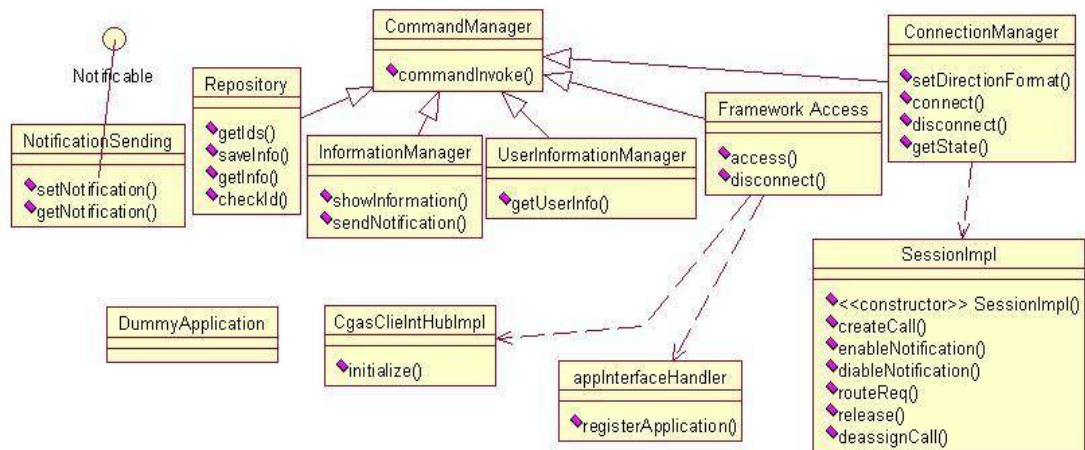


7.3 RECOMENDACIONES REALIZADAS AL NIVEL UNO

Sobre el nivel 1 no se realizaron recomendaciones, se presentan a continuación los diagramas de clases y secuencia de este nivel.

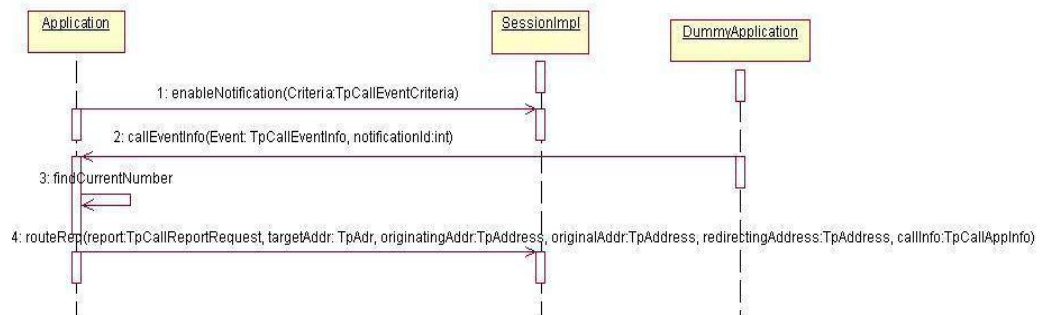
7.3.1 Diagrama de clases aplicación de prueba. Nivel 1. A continuación se muestra el diagrama de clases, del primer nivel.

Figura 52. Diagrama de Clases Nivel 1



El diagrama de secuencia de la aplicación de prueba, usando el nivel 1 es el siguiente:

Figura 6. Diagrama de secuencia nivel 1. Implementación aplicación de prueba



8. CONCLUSIONES

Para el diseño e implementación del servicio propuesto se utilizó como base teórica el trabajo “Ambiente de desarrollo para la implementación de nuevos servicios telemáticos sobre la red multi-servicios de EMCALI”. La información presentada en ese trabajo permitió acelerar el proceso de desarrollo del servicio, esta información compensó la falta de documentación de EMCALI acerca de su plataforma de aplicaciones. Los resultados obtenidos en este proyecto son una prueba de la validez de ese ambiente desarrollado para el caso de la red multiservicios de EMCALI.

Este servicio se desarrolló de forma simultánea con el modelo del GITI, el objetivo de esta metodología fue enriquecer el modelo con experiencias prácticas y comprobarlo con resultados prácticos. De esta forma a medida que se avanzaba en el desarrollo del servicio se afinaba la propuesta presentada en el modelo. Las observaciones mostradas en este documento son un resumen de las recomendaciones que se presentaron al modelo a medida que se evaluaba el proceso de desarrollo del servicio. Los resultados finales obtenidos demuestran que el modelo propuesto es válido y aplicable a la arquitectura de la plataforma de aplicaciones de EMCALI.

Para el desarrollo de aplicaciones utilizando el API de PCH, que es una abstracción de OSA/Parlay, no es necesario un amplio conocimiento sobre los protocolos de señalización en telecomunicaciones, esta facilidad permite el desarrollo rápido de aplicaciones y servicios.

La falta de documentación e información acerca de PCH y la plataforma fue un problema recurrente que dilató los tiempos estipulados para el desarrollo de la aplicación. Para un adecuado desarrollo de servicios es necesario contar con documentación suficiente que permita aprovechar las facilidades que brinda OSA/Parlay.

Para el registro y activación de las aplicaciones debe utilizarse la consola de mantenimiento y administración, PCH-OAM. Esta aplicación es el mecanismo offline disponible para autorizar a las aplicaciones el acceso a determinados servicios. Sin esta aplicación no es posible el despliegue de nuevos servicios o aplicaciones de forma segura.

Con el estado actual de la red multiservicios de EMCALI y su plataforma de aplicaciones es posible desarrollar nuevas aplicaciones y servicios telemáticos utilizando Java como lenguaje de programación y el API de PCH. Sin embargo, es necesario contar con una versión completa y habilitada de la aplicación PCH-OAM para realizar una correcta configuración de las aplicaciones.

BIBLIOGRAFIA

- [1] BERRUTO, Ermanno. Evolución hacia la NGN. Milan, Giugno 13 de 2007. ANFoV – Observatorio NGN. [en línea] [Consultado 20 de Junio de 2008]
Disponible en Internet:
http://www.anfov.it/s_leNostreAttivita/docs_osservatori/NGN_presentazioneWind_13_6_07.pdf
- [2] CHNEPS-SCHNEPPE Manfred y UMITRY AbavaNet. Parlay Architecture and NGN [en línea] [Consultado 18 de Junio de 2008] Disponible en internet
http://www.abavanet.ru/download/1-4-3_schneps.pdf
- [3] COHEN, Tracy y HERNEDEZ, Janet [en línea]. El camino hacia las redes de próxima generación-pioneros [en línea]. [Consultado 24 de Junio de 2008]
Disponible en Internet: <http://www.itu.int/itu-news/manager/display.asp?lang=es&year=2007&issue=02&ipage=next-generation2&ext=html#top>
- [4] Departamento de Planeación e Ingeniería EMCALI. Marco Teórico RMS EMCALI 1. Santiago de Cali. 2007
- [5] Ericsson. Ericsson Network Resource Gateway Software Development Kit User Guide. [en línea] [Consultado 20 de Junio de 2008]
Disponible en Internet: <http://archive.ericsson.net/service/internet/picov/get?DocNo=28701-FGC101805&Lang=EN&Rev=A>
- [6] FALCARIN Paolo y LICCIARDI Carlos Alberto. Análisis de tecnologías para el desarrollo de servicios en NGN. [en línea]
Disponible en Internet <http://softeng.polito.it/falcarin/docs/Annals03.pdf>
- [7] FALCARIN, Paolo, LICCIARDI Carlos Alberto y ANDREETTO Alessandra. Service Opportunities for Next Generation Networks [en línea]
[Consultado 20 de Junio de 2008] Disponible en internet:
http://phoenix.labri.fr/documentation/sip/Documentation/Papers/Programming_SIP/Paper_Publication_and_Draft/V02N01Art263.pdf
- [8] GUEVARA, Diana Marcela. Ambientes de desarrollo para la implementación de nuevos servicios telemáticos sobre la red multi-servicios de EMCALI. Pasantía para Ingeniera Electrónica Santiago de Cali: Universidad Autónoma de Occidente. Facultad de Ingeniería. 2008. 206p
- [9] IBM Rational Unified Process. [en línea] [Consultado 21 de Junio de 2008]
Disponible en Internet
ftp://ftp.software.ibm.com/software/rational/web/datasheets/RUP_DS.pdf
- [10] Infor-Communications Development Authority of Singapore. “Next Generation Networks (NGN) Technical Framework”. Singapore Febrero de

2007. [en línea] [Consultado 24 de Junio de 2008]

Disponible en internet http://www.ida.gov.sg/doc/Policies%20and%20Regulation/Policies_and_Regulation_Level2/20060424161505/IDARSNGN_TechFrw.pdf

[11] JAIN, Ravi. BAKKER, John-Luc. ANJUUM Farooq. Programming Converged Networks. Wiley 2005. 268 p

[12] MONDRAGON Oscar, SOLARTE Zeida. Modelo Arquitectónico para la implementación de nuevos servicios telemáticos sobre la red multiservicios de EMCALI. Santiago de Cali, 2008. Universidad Autónoma de Occidente

[13] MUÑOZ O Mario. SCMM Metamodelo para la creación de aplicaciones en Redes de Siguiete Generación. Tesis doctoral para Ingeniero en Comunicación. Leganés: Universidad de Carlos III de Madrid. 2003. 276p

[14] Net4Call Press Release. Octubre de 2005. [en línea]

[Consultado 22 de Junio de 2008]

Disponible en Internet:

http://www.parlay.org/imwp/idms/popups/pop_download.asp?contentID=7019

[15] REHEMTULLA, Mebs y HUGHES Glenn. Service creation: Meeting the product lifecycle challenge. Enero de 2006. IBM Product Lifecycle management. [en línea] [Consultado 20 de Junio de 2008]

Disponible en internet:

[http://www-](http://www-03.ibm.com/industries/telecom/doc/content/bin/Service_Creation_Meeting_the_product_lifecycle_challenge.pdf)

[03.ibm.com/industries/telecom/doc/content/bin/Service_Creation_Meeting_the_product_lifecycle_challenge.pdf](http://www-03.ibm.com/industries/telecom/doc/content/bin/Service_Creation_Meeting_the_product_lifecycle_challenge.pdf)

[16] UNMEHOPA Musa, VEMURI Kumar, BENNET Andy. Parlay/OSA: From Standards to Reality. Willey 2006. 295 p

[17] VALLECILLA, Víctor Manuel y CARMONA, Marcela. Diseño e implementación de servicios telemáticos sobre la red inteligente NGN de EMCALI. Pasantía para Ingeniería informática .Santiago de Cali: Universidad Autónoma de Occidente. Facultad de ingeniería. Departamento de sistemas de información. 2007. 256p

ANEXOS

Anexo A. Acta de reunión Octubre 3 de 2008.

Fecha: Octubre 3 de 2008

Tema: Captura de requerimientos y compromisos iniciales

Asistentes:

Luis Fabián Troyano
EMCALI. Área Comercial

Eugenio Castro
EMCALI. Planeación

Diana Guevara
EMCALI. Red Multiservicios

Hugo Murillo
EMCALI. Red Multiservicios

Juan Carlos Vita
EMCALI. Red Multiservicios

Andrés Murillo
Pasante en investigación UAO
Oscar Mondragón
Director de proyecto de grado - UAO

Conclusiones y compromisos de la reunión

- Se acepta el servicio de Portabilidad Numérica como ejercicio didáctico para conocer de forma práctica las herramientas de OSA/Parlay y la plataforma de aplicaciones de la red multiservicios de EMCALI
- Pablo Cardona Ingeniero de ZTE atenderá las inquietudes y solicitudes relacionadas con los productos de ZTE.
- Para la implementación y prueba se utilizará la plataforma conocida como Parlay Gateway 2 que no tiene corriendo ninguna aplicación actualmente.

Anexo B. Acta de reunión Marzo 9 de 2009

Fecha: Marzo 9 de 2009

Tema: Acceso a la base de datos de la plataforma de aplicaciones para finalizar configuración de la aplicación

Asistentes:

Luis Fabián Troyano
EMCALI. Área Comercial

Zeida Solarte
Directora Programa Ingeniería electrónica . Universidad Autónoma de Occidente

Andrés Murillo
Pasante en investigación. Universidad autónoma de Occidente

Oscar Mondragón
Director de proyecto de grado. Universidad Autónoma de Occidente

Conclusiones y compromisos de la reunión

- Según el análisis realizado sobre la plataforma de aplicaciones y PCH, una forma de finalizar la configuración de la aplicación es accediendo a la base de datos de la plataforma. Sin embargo, debido a razones contractuales ZTE debe otorgar autorización para este acceso.
- Se programa una reunión con Pablo Cardona de ZTE para realizar esta solicitud.

Anexo C. Acta de reunión Marzo 11 de 2009

Fecha: Marzo 11 de 2009

Tema: Acceso a la base de datos de la plataforma de aplicaciones para finalizar configuración de la aplicación

Asistentes:

Luis Fabián Troyano
EMCALI. Área Comercial

David Blandón
EMCALI. Ingeniería

Pablo Cardona
ZTE Corporation. Departamento de Post Venta

Zeida Solarte
Directora Programa Ingeniería electrónica . Universidad Autónoma de Occidente

Andrés Murillo
Pasante en investigación. Universidad autónoma de Occidente

Oscar Mondragón
Director de proyecto de grado. Universidad Autónoma de Occidente

Conclusiones y compromisos de la reunión

- Gracias al trabajo y análisis demostrado hasta el momento se le ha demostrado a ZTE que a pesar de que PCH posee información propietaria, también hay mucha información que está definida en un estándar abierto, por esta razón EMCALI puede solicitar este tipo de información.
- El servicio en desarrollo será utilizado para validar las capacidades de la plataforma de aplicaciones de la red multiservicios, especialmente en el área de desarrollo de aplicaciones por terceros.
- Para que ZTE le entregue el acceso a la base de datos zxinsys a EMCALI debe solucionarse un tema contraactual entre las dos compañías. Una alternativa para finalizar la configuración de la aplicación es utilizar la consola de mantenimiento y administración PCH-OAM

Anexo D. Acta de reunión Abril 17 de 2009

Fecha: Abril 17 de 2009

Tema: Solicitud de versión completa consola de mantenimiento PCH-OAM

Asistentes:

David Blandón

EMCALI. Ingeniería

Pablo Cardona

ZTE. Departamento de Post Venta

Zeida Solarte

Directora Programa Ingeniería electrónica . Universidad Autónoma de Occidente

Andrés Murillo

Pasante en investigación. Universidad autónoma de Occidente

Oscar Mondragón

Director de proyecto de grado. Universidad Autónoma de Occidente

Conclusiones y compromisos de la reunión

- La versión que posee EMCALI actualmente del PCH-OAM no tiene todas las opciones habilitadas, entre las opciones deshabilitadas se encuentra la de creación de notificaciones.
- Pablo Cardona realizará la solicitud a ZTE China para que le envíen una versión completa del PCH-OAM.
- Los resultados obtenidos por la implementación del servicio de Portabilidad Numérica servirán para validar las capacidades de la plataforma de aplicaciones de la NGN de EMCALI. Si es posible implementar en su totalidad este servicio, entonces la plataforma es apta para el desarrollo de aplicaciones por terceros.
- Se revisará la documentación disponible acerca del Parlay Client Hub (PCH) y se realizará una petición a ZTE para complementar esta documentación.

Anexo E. Acta de reunión Marzo 12 de 2009

Fecha: Marzo 12 de 2009

Tema: Acceso a la base de datos de la plataforma de aplicaciones para finalizar configuración de la aplicación

Asistentes:

Luis Fabián Troyano
EMCALI. Área Comercial
David Blandón
EMCALI. Ingeniería
Diana Guevara
EMCALI. Red Multiservicios

Pablo Cardona
ZTE Corporation. Departamento de Post Venta

Zeida Solarte
Directora Programa Ingeniería electrónica . Universidad Autónoma de Occidente
Andrés Murillo
Pasante en investigación. Universidad autónoma de Occidente
Oscar Mondragón
Director de proyecto de grado. Universidad Autónoma de Occidente

Conclusiones y compromisos de la reunión

- Como alternativa para solucionar el acceso a la base de datos se utilizará la consola de mantenimiento y administración PCH-OAM.

Anexo F . Modelo para el desarrollo de servicios en la red NGN de emcali – telecomunicaciones

MODEL FOR SERVICES DEVELOPMENT IN EMCALI-TELECOMUNICACIONES NGN NETWORK

Oscar Mondragon
Master en Wireless Systems and Related
Technologies
Docente Tiempo Completo
Universidad Autónoma de Occidente
ohmondragon@uao.edu.co

Zeida Solarte
Especialista en Redes y Servicios Telemáticos
Directora
Programa de Ingeniería Electrónica
Universidad Autónoma de Occidente
zsolarte@uao.edu.co

RESUMEN

La tendencia actual en el mundo de las telecomunicaciones a integrar los servicios en una única infraestructura de red, ha propiciado la aparición de los modelos de Red de Nueva Generación o NGN. Las Empresas Municipales de Cali - EMCALI, acordes con esta tendencia han adquirido recientemente una plataforma de red multiservicio ZTE que permite la oferta de servicios integrados de voz, datos y video, con altos estándares de calidad y que soporta la arquitectura de OSA/Parlay para la construcción de servicios. Para hacer eficiente la creación de servicios sobre esta plataforma es necesario crear un modelo arquitectónico que les permita el desarrollo rápido de aplicaciones telemáticas, ya sea por la misma EMCALI o por terceros, que puedan ser fácilmente desplegados sobre su infraestructura de red y que le permita mejorar la oferta de servicios a los usuarios finales y la calidad con la que se prestan estos.

Palabras Claves: Red de Nueva Generación, OSA/Parlay, Servicios Telemáticos, Modelo de Desarrollo, EMCALI.

ABSTRACT

The current trend in the world of telecommunications to integrate services into a single network infrastructure, has led to the emergence of next generations networks models or NGN. In line with this trend, the Empresas Municipales de Cali – EMCALI, has recently acquired a ZTE multiservice network platform, that allows supply services of integrated voice, data and video, with high quality of services and that supports the OSA/Parlay architecture for service construction. To make efficient building services on this platform is necessary to create an architectural model that allows the rapid development of telematic applications, either by itself or by third parties, in this form, the services can be easily deployed over the network infrastructure and enabling it to improve the supply of services to end users and the quality with the services are provided.

Key Words: Next Generation Network, OSA/Parlay, Telematic Services, Development Model, EMCALI

MODELO PARA EL DESARROLLO DE SERVICIOS

El Modelo para la Creación de Servicios en la red NGN de Emcali MCSE, está basado en el

metamodelo SCMM (*Service Creation MetaModel*) [7]. SCMM es un marco genérico de referencia del cual se pueden extender modelos específicos para los diferentes entornos que formarán parte de las redes de siguiente generación. Esta extensión se realiza utilizando el concepto de herencia.

Basados en el metamodelo SCMM, se definen 3 capas de referencia para el modelo MCSE como se muestra en la Figura 1.



Figura 1. Estructura del modelo MCSE

La capa de servicios es la que especifica los procedimientos para acceder a las APIs que definen las capacidades de servicio que provee PCH.

De acuerdo al metamodelo trabajado, MCSE define tres niveles jerárquicos servicios, estos niveles son:

- **Nivel 1.** En este nivel se tendrán las capacidades básicas aplicables a todos los servicios y por lo tanto constituirán el nivel jerárquico más alto. Este nivel se basa en la definición de las capacidades de enviar eventos e invocar comandos.
- **Nivel 2.** Se recogen en este nivel algunas capacidades que serán comunes a grupos de servicios. Se clasifican los servicios en familias y se definen interfaces genéricas para cada tipo de ellos. Lógicamente, las capacidades definidas a nivel 2 pueden heredar de las definidas a nivel 1 siguiendo el mecanismo de herencia.
- **Nivel 3.** En este nivel se definirán interfaces específicas para las capacidades definidas por PCH.

En la capa *descripción de servicios* se define un mecanismo se posibilita el anuncio y aprendizaje mediante un protocolo de descubrimiento de servicios de las capacidades soportadas

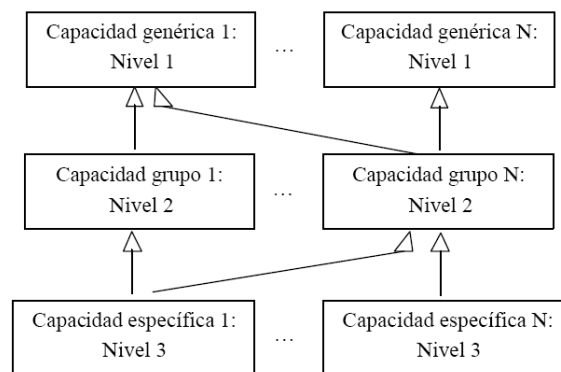
por cada uno de los servicios disponibles en un determinado entorno y en un determinado momento. Estos protocolos dinámicos de descubrimiento de servicios necesitan el soporte de un lenguaje con el que poder describir los servicios que se exportan a través de interfaces abiertos. SCMM define un lenguaje basado en XML [8] para la descripción de servicios denominado GSDL (*Generic Service Description Language*), el cual será adoptado por el modelo MCSE.

En la capa de aplicación se especifica mecanismos sencillos, flexibles, ubicuos y escalables de creación de servicios adaptados a los entornos de redes de siguiente generación. Para ellos se realiza una especificación de las aplicaciones en forma genérica e independiente de los dispositivos donde se ejecutarán y se mantendrán los detalles específicos de cada entorno escondidos dentro del entorno de ejecución. Para ello, dentro del modelo MCSE se ha adoptado la realización de esta descripción mediante un lenguaje basado en XML denominado GSCL (*Generic Service Creation Language*).

3.1. Servicios

En esta capa se define una descripción jerárquica de las capacidades de servicio aplicadas a PCH que permitirá diseñar entornos genéricos y escalables. En la Figura 9. se detallan los tres niveles jerárquicos.

Figura 9. Niveles jerárquicos de las capacidades del servicio



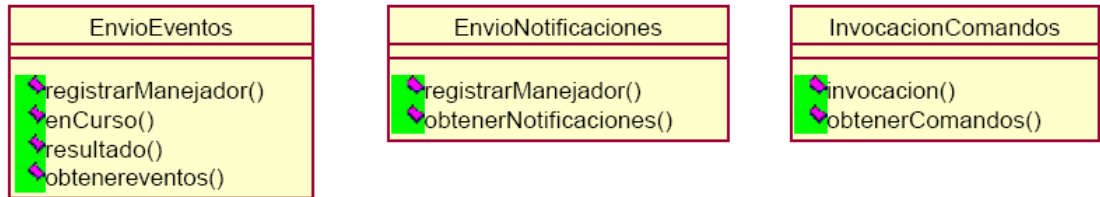
3.1.1. Servicios Nivel 1

El primer nivel en la jerarquía de servicios lo constituyen las capacidades básicas de interacción de éstas con las aplicaciones. Básicamente un servicio puede por una parte exportar funcionalidades que podrán ser invocadas remotamente desde las aplicaciones y por otra mandar notificaciones y eventos a dichas aplicaciones ante determinadas circunstancias. Estas

capacidades básicas son genéricas a todo servicio.

En la Figura 10. se muestra el diagrama de clases para este nivel.

Figura 10. Diagrama de Clases del Nivel 1



La capacidad de Invocación de Comandos permite a las aplicaciones la interacción con servicios mediante la invocación de comandos. El método de invocación de comandos recibirá como parámetros el nombre del comando a invocar y los argumentos y devolverá el resultado de la invocación del comando. Este servicio se puede considerar como una pasarela que permite la invocación genérica de comandos desde las aplicaciones. Por su parte el segundo método del servicio permite obtener los comandos soportados. De esta manera, una aplicación genérica que hiciera uso de las capacidades de servicio a nivel 1 podría obtener las funcionalidades asociadas a todo servicio que derive del servicio de invocación de comandos y realizar la llamada a cualquiera de los métodos asociados.

La segunda capacidad de servicio es la capacidad de enviar notificaciones a las aplicaciones. Esta capacidad permite que las aplicaciones registren una referencia remota en el servicio. Esta referencia remota contiene la información de la dirección de red del dispositivo que ejecuta la aplicación así como el puerto y método a invocar cuando se produce la notificación. Cuando se produce la notificación se invocará el método remoto que se ha registrado en la capacidad de envío de notificaciones. Este método requerirá como parámetro un *String* con la información propia del evento que se ha producido.

El envío de notificaciones posibilita que el GW le envíe notificaciones acerca de eventos relevantes a la aplicación. A través de esta capacidad pueden implementarse servicios orientados a eventos.

La última de las capacidades de servicio es la capacidad de enviar eventos cuya funcionalidad básica es similar a la capacidad anterior pero se le añade la posibilidad de que las aplicaciones tomen acciones como respuesta a la señalización del evento. Así pues, el servicio de envío de eventos permite que las aplicaciones registren un manejador del evento. Por otra parte permite que las aplicaciones respondan al evento mediante un par de métodos nuevos, el método *encurso()* que será invocado por las aplicaciones si vencido un temporizador todavía no están en condiciones de responder al evento para solicitar más tiempo de proceso al servicio, y el método que invocarán las aplicaciones como respuesta al evento.

3.1.2. Servicios Nivel 2

El segundo nivel jerárquico pretende realizar un modelado por familias entre los tipos de capacidades con las que cuenta PCH. Por lo tanto, el segundo nivel jerárquico pretende crear interfaces con funcionalidad común a servicios de la misma familia. Esto permitirá una cierta interactividad con todo servicio sin necesidad de un conocimiento exhaustivo del mismo simplemente con el conocimiento de la familia a la que pertenece. Los tipos de capacidades definidos para este nivel son:

- Capacidad de presentar información a un usuario.
- Capacidad de captar información de un usuario.
- Capacidad de establecer conexiones con dispositivos remotos.
- Capacidad de enviar información a dispositivos remotos.
- Capacidad de hacer de repositorio de información.

En la Figura 11. Se muestra el diagrama de clases de Nivel 2.

Figura 11. Diagrama de clases de nivel 2



La capacidad de mostrar información permite mostrar información a usuarios de forma genérica con independencia del formato de presentación. El único método del que consta recibe como argumento el mensaje a presentar. Un dispositivo que exporte esta capacidad de servicio permitirá a aplicaciones remotas interactuar con los usuarios que accedan a dicho dispositivo.

La capacidad de obtener información de usuario permite obtener información de usuario completando la interacción con el usuario que se inició con la capacidad anterior.

La capacidad de establecimiento de conexiones remotas permite modelar el establecimiento y liberación de conexiones remotas. Esta capacidad de servicio se relaciona directamente con la creación de servicios en los entornos tradicionales de redes de telecomunicación. Básicamente se tiene la capacidad de establecer conexiones, consultar su estado y liberar dichas conexiones. Adicionalmente y para aquellas capacidades que permitan establecer conexiones de acuerdo con diferentes formatos de numeración se puede establecer dicho formato para la invocación del establecimiento de nuevas conexiones.

El envío de información a través de conexiones establecidas según la capacidad de servicio anterior se realiza mediante la capacidad de servicio de envío de información remota.

La capacidad de hacer de repositorio de información permite a las aplicaciones remotas realizar consultas de información. La forma en la que la capacidad de servicio de repositorio obtiene la información puede ser variada. Por una parte se puede permitir a las aplicaciones que guarden información para su posterior recuperación o para la recuperación por parte de otras aplicaciones o por otra parte, la alimentación de los datos en el repositorio puede hacerse por

otros mecanismos ajenos a la comunicación entre servicios y aplicaciones.

3.1.3. Servicios Nivel 3

El tercer nivel jerárquico contiene el detalle de cada una de las funcionalidades de PCH. El hecho de que estas capacidades hereden de capacidades de nivel 2 que a su vez heredan de capacidades de nivel 1 permite a las aplicaciones que no conocen previamente el tipo de funcionalidad ofrecida a nivel 3 el poder hacer un uso limitado de él mediante el conocimiento que tienen de los niveles superiores.

Las capacidades de nivel 3 de PCH pueden agruparse de acuerdo a la funcionalidad de la siguiente manera:

- Manejo de sesiones.
- Manejo de aplicaciones.
- Pool de hilos.
- Conexión a la base de datos.
- Pool de números de terminal.
- La función log.

3.2. DESCRIPCIÓN DE SERVICIOS

La capa de *descripción de servicios* permite que aplicaciones desarrolladas con base en servicios web [21] puedan hacer uso de las capacidades ofrecidas por la plataforma.

Para permitir propagar la información de los servicios soportados en cada dispositivo de forma dinámica se utiliza el lenguaje de marcado denominado GSDL (*Generic Service Description Language*). Este lenguaje facilita el descubrimiento y utilización de los servicios por parte de las aplicaciones.

GSDL esta definido utilizando XMLSchema [10] y utiliza el modelado jerárquico para las capacidades de servicio. GSDL es un complemento de WSDL que permite incorporar a la

definición de los servicios Web la información de herencia. Un anuncio de un servicio específico de nivel 3 en GSDL incorpora información de los servicios de nivel 1 y de nivel 2 de los que dicho servicio de nivel 3 hereda.

3.3. CAPA DE APLICACIONES

Para la creación de servicios se usa el lenguaje GSCL basado en XML que describe la secuencia de acciones que se toman en el servicio. Básicamente se representa el servicio como una máquina de estados: Estados, eventos activadores del estado y comandos generados. Esta estructura permite crear dinámicamente servicios que luego serán convertidos en código Java.

Las etiquetas permitidas por el lenguaje para describir los servicios son

CallFlow: Representa el flujo del servicio. Contiene la máquina de estados del servicio.

Agents: Colección de agentes de usuario que participan en el servicio

Agent: Agentes de usuario

StateMachine: Representa la máquina de estados del servicio

StateGroup: Grupo de estados. Es útil cuando un estado contiene varios estados internos.

State: Representa un estado de la máquina de estados.

TriggerEvent: Representa un evento que inicializa un estado específico.

AcceptableEvent: Representa un evento que es válido cuando el servicio se encuentra en un estado específico.

Timer: Útil para representar tiempos entre diferentes comandos ejecutados dentro de un estado.

Generate: Colección de comandos que se generan en un estado

Command: Representa un comando que se genera en un estado específico.

Parameter: Representa un parámetro de un comando

Referencias

- [1] Migración a NGN. Centro de Investigación en Telecomunicaciones CINTEL, 2003.

Disponible en Internet:

<http://www.cintel.org.co:8080/cintel/opencms/cintel/inicio/lineas/linea/migracion.html>.

- [2] POPOFF, Jeff. Parlay/OSA. Boston: The Parlay Group, 2005. Disponible en Internet: <http://www.parlay.org>
- [3] EMCALI EICE ESP. Descripción de la Red Existente. Santiago de Cali, 2008.
- [4] ZTE. Propuesta técnica plataforma de servicios. Santiago de Cali, 2005.
- [5] ZTE. Plataforma Unificada de Servicios ZXUP10. Santiago de Cali, 2005.
- [6] ZTE. Introduction to Parlay Client Hub (PCH). Santiago de Cali, 2008.
- [7] MUÑOZ O. Mario. Tesis Doctoral SCMM: Metamodelo para la Creación de Aplicaciones en Redes de Siguiete Generación. Universidad Carlos III de Madrid. Doctorado en Tecnologías de las Comunicaciones. 2003
- [8] XML Extensible Markup Language. W3C. Disponible en: <http://www.w3.org/XML/>.
- [9] Web Services Activity. W3C. Disponible en: <http://www.w3.org/2002/ws/>
- [10] XMLSchema. W3C. Disponible en: <http://www.w3.org/XML/Schema>